WILLKIE FARR & GALLAGHER LLP

1875 K Street, NW
Washington, DC 20006

Tel: 202 303 1000
Fax: 202 303 2000

March 22, 2006

Marlene H. Dortch
Federal Communications Commission
Office of the Secretary
445 12th Street, SW
Washington, DC 20554

Re: In re Telecommunications Relay Services and Speech-to-Speech Services for Individuals with Hearing and Speech Disabilities; Petition for Declaratory Ruling on Video Relay Service Interoperability, CG Docket No. 03-123

Dear Ms. Dortch:

On March 20, 2006, representatives of Snap Telecommunications, Inc. ("Snap"), Aequus Technologies Corp. ("Aequus"), and WorldGate Communications, Inc. ("WorldGate") met with Tom Chandler, Chief, Disability Rights Office; Jay Keithley, Deputy Bureau Chief, Consumer & Governmental Affairs Bureau; Greg Hlibok, Disability Rights Office; and Sharon Diskin, Office of the General Counsel. Also attending the meeting were David Dinin, President, Aequus; Daryl Crouse, President and Founder, Snap; Randy Gort, General Counsel, WorldGate; Richard Westerfer, Chief Operating Officer and Senior Vice-President, WorldGate; and the undersigned.

During the meeting, the parties supported the petition in the above-captioned proceeding and opposed restrictive marketing practices, IP blocking, or other techniques by a VRS provider designed to prevent a hearing-impaired individual from placing a VRS call to a different VRS provider. However, the parties also cautioned the Commission to avoid the adoption of requirements in this proceeding that could have the inadvertent effect of impeding the ability of new or existing VRS providers from introducing VRS equipment and services implementing the newer and more robust open standard called Session Initiation Protocol ("SIP"). The parties described the benefits of SIP in providing functionally equivalent service for the hearing-impaired community, the fact that SIP is increasingly being embraced as the standard of choice in the video phone and VoIP arenas, and that SIP is the focus of significant efforts by various industry players to establish E-911 solutions for VoIP and VRS. The attached materials were handed out to the Commission participants at the end of the meeting.

Should you have any questions regarding this matter, please do not hesitate to contact the undersigned.

Respectfully submitted,

/s/

Francis M. Buono
Counsel for Snap Telecommunications, Inc. and Aequus Technologies Corp.

Attachments

cc: Thomas Chandler (without attachments)
Jay Keithley (without attachments)
Greg Hlibok (without attachments)
Sharon Diskin (without attachments)

A Comparison of SIP and H.323 for Internet Telephony

Henning Schulzrinne
Dept. of Computer Science, Columbia University
New York, NY 10027
hgs@cs.columbia.edu

Jonathan Rosenberg
Bell Laboratories
Holmdel, NJ 07733
jdrosen@bell-labs.com

Abstract—**Two standards have recently emerged for signaling and control for Internet Telephony. One is ITU Recommendation H.323, and the other is the IETF Session Initiation Protocol (SIP). These two protocols represent very different approaches to the same problem: H.323 embraces the more traditional circuit-switched approach to signaling based on the ISDN Q.931 protocol and earlier H-series recommendations, and SIP favors the more lightweight Internet approach based on HTTP. In this paper, we compare SIP and H.323 on complexity, extensibility, scalability, and features.**

I. Introduction

In order to provide useful services, Internet telephony requires a set of control protocols for connection establishment, capabilities exchange, and conference control. Currently, two protocols exist to meet this need. One is ITU-T H.323, and the other is the IETF Session Initiation Protocol (SIP). In this paper, we compare the two protocols on complexity, extensibility, scalability, and services.

The ITU H.323 series of recommendations ("Packet Based Multimedia Communications Systems") defines protocols and procedures for multimedia communications on, among other things, the Internet. It includes H.245 for control, H.225.0 for connection establishment, H.332 for large conferences, H.450.1 H.450.2 and H.450.3 for supplementary services, H.235 for security, and H.246 for interoperability with circuit-switched services. H.323 started out as a protocol for multimedia communication on a LAN segment without QoS guarantees, but has evolved to try and fit the more complex needs of Internet telephony.

H.323 is based heavily on the ITU multimedia protocols which preceded it, including H.320 for ISDN, H.321 for B-ISDN, and H.324 for GSTN terminals. The encoding mechanisms, protocol fields, and basic operation are somewhat simplified versions of the Q.931 ISDN signaling protocol.

The Session Initiation Protocol (SIP) [1], developed in the MMUSIC working group of the IETF, takes a different approach to Internet telephony signaling by reusing many of the header fields, encoding rules, error codes, and authentication mechanisms of HTTP.

In both cases, multimedia data will likely be exchanged via RTP, so that the choice of protocol suite does not influence Internet telephony QOS.

II. Complexity

H.323 is a rather complex protocol. The sum total of the base specifications alone (not including ASN.1 and PER) is 736 pages. SIP, on the other hand, along with its call control extensions and session description protocols totals merely 128 pages. H.323 defines hundreds of elements, while SIP has only 37 headers (32 in the base specification, 5 in the call control extensions), each with a small number of values and parameters, but that contain more information. A basic, but interoperable SIP Internet telephony implementation can get by with four headers (To, From, Call-ID, and CSeq) and three request types (INVITE, ACK, and BYE) and is small enough to be assigned as a homework programming problem. A fully functional SIP client agent, with a graphical user interface, has been implemented in just two man-months.

H.323 uses a binary representation for its messages, based on ASN.1 and the packed encoding rules (PER). ASN.1 generally requires special code-generators to parse. SIP, on the other hand, encodes its messages as text, similar to HTTP [2] and the Real Time Streaming Protocol (RTSP) [3]. This leads to simple parsing and generation, particularly when done with powerful text processing languages such as Perl. The textual encoding also simplifies debugging, allowing manual entry and perusing of messages. Its similarity to HTTP also allows for code-reuse; existing HTTP parsers can be quickly modified for SIP usage.

H.323's complexity also stems from its use of several protocol components. There is no clean separation of these components; many services require interactions between several of them. (Call forward, for example, requires components of H.450, H.225.0, and H.245.) The use of several different protocols also complicates firewall traversal. Firewalls must act as application level proxies [4], parsing the entire message to arrive at the required fields. The operation is stateful since several messages are involved in call setup. SIP, on the other hand, uses a single request that contains all necessary information.

H.323 also provides for an array of options and methods for accomplishing a single task. For example, there are three distinct ways in which H.245 and H.225.0 may be used together: the original H.323v1 approach of separate connections, H.245 tunneling through H.225.0, and FastStart in H.323v2. In the original approach, the call signaling channel is set up first, the H.245 control channel is established, and finally the media channels are opened. This can require many round trips for call setup. FastStart includes the media channel information in the original call invitation, avoiding the need to open the H.245 channel. In H.245 tunnelling, the H.245 channel is still used, but its messages are carried over the call signaling channel. Even though FastStart is much more efficient, H.323 allows any of the three and thus, firewalls, end systems, gatekeepers, and gateways must support all of them. As with any protocol, large option spaces lead to feature interaction and the need for profiles. (How does encryption of the H.245 channel work when its tunneled through H.225.0, for example?).

An additional aspect of H.323's complexity is its duplication of some of the functionality present in other parts of the protocol. In particular, H.323 makes use of RTP and RTCP. RTCP has been engineered to provide various feedback and conference control functions in a manner which scales from two-party conferences to thousand-party broadcast sessions. H.245, however, provides its own mechanisms for both feedback and simple conference control (such as obtaining the list of conference participants). These H.245 mechanisms are redundant, and have been engineered for small to medium-sized conferences only.

III. Extensibility

Extensibility is a key metric for measuring an IP telephony signaling protocol. Telephony is a tremendously popular, critical service, and Internet telephony is poised to supplant the existing circuit switched infrastructure developed to support it. As with any heavily used service, the features provided evolve over time as new applications are developed. This makes compatibility among versions a complex issue. As the Internet is an open, distributed, and evolving entity, one can expect extensions to IP telephony protocols to be widespread and uncoordinated. This makes it critical to build in powerful extension mechanisms from the outset.

SIP has learned the lessons of HTTP and SMTP (both of which are widely used protocols that have evolved over time), and built in a rich set of extensibility and compatibility functions. By default, unknown headers and values are ignored. Using the Require header, clients can indicate named feature sets that the server must understand. When a request arrives at a server, it checks the list of named features in the Requires header. If any of them are not supported, the server returns an error code and lists the set of features it does understand. The client can then determine the problematic feature and fall back to simpler operation. The feature names are based on a hierarchical namespace, and new feature names can be registered with IANA. This means that any developer can create new features in SIP, and then simply register a name for them. Compatibility is still maintained across different versions.

To further enhance extensibility, numerical error codes are hierarchically organized, as in HTTP. There are six basic classes, each of which is identified by the hundreds digit in the response code. Basic protocol operation is dictated solely by the class, and terminals need only understand the class of the response. The other digits provide additional information, usually useful but not critical. This allows for additional features to be added by defining semantics for the error codes in a class, while achieving compatibility.

The textual encoding means that header fields are self-describing. It is self-evident what the meaning of the To, From, and Subject fields are. As new header fields are added in various different implementations, developers in other corporations can determine usage just from the name, and add support for the field. This kind of distributed, documentation-less standardization has been common in the Simple Mail Transfer Protocol (SMTP), which has evolved tremendously over the years.

As SIP is similar to HTTP, mechanisms being developed for HTTP extensibility can also be used in SIP. Among these are the Protocol Extensions Protocol (PEP), which contains pointers to the documentation for various features within the HTTP messages themselves.

H.323 provides extensibility mechanisms as well. These are generally nonstandardParam fields placed in various locations in the ASN.1. These params contain a vendor code, followed by an opaque value which has meaning only for that vendor. This does allow for different vendors to develop their own extensions. However, it has some limitations. First, extensions are limited only to those places where a non-standard parameter has been added. If a vendor wishes to add a new value to some existing parameter, and there is no placeholder for a nonstandard element, one cannot be added. Secondly, H.323 has no mechanisms for allowing terminals to exchange information about which extensions each supports. As the values in non-standard parameters are not self-describing, this limits interoperability among terminals from different manufacturers.

In addition, H.323 requires full backwards compatibility from each version to the next. As various features come and go, the size of the encodings will only increase. However, SIP allows for older headers and features to gradually disappear as they are no longer needed, keeping the protocol and its encoding clean and concise.

A critical issue for extensibility are audio and video codecs. There are hundreds of codecs that have been developed, many of which are proprietary. SIP uses the Session Description Protocol (SDP) to convey the codecs supported by an endpoint in a session. Codecs are identified by string names, which can be registered by any person or group with IANA, and then used. This means that SIP can work with any codec, and other implementations can determine the name of the codec, and contact information for it, from IANA.

In H.323, each codec must be centrally registered and standardized. Currently, only ITU developed codecs have codepoints. As many of these carry significant intellectual property, there is no free, sub-28.8 kb/s codec which can be used in an H.323 system. This presents a significant barrier to entry for small players and universities.

Furthermore, SIP allows for new services to be defined through a few powerful third-party call control mechanisms. These mechanisms allow a third party to instruct another entity to create and destroy calls to other entities. As the controlled party executes the instructions, status messages are passed back to the controller. This allows the controller to take further actions based on some local program execution. This is much like the IN model in traditional telephony. As there are hundreds of telephony services currently defined, it is unreasonable to attempt to write specifications for each. SIP allows these services to be deployed by basing them on simple, standardized mechanisms. These mechanisms can be used to construct a variety of services, including blind transfer, operator assisted transfer, three-party calling, bridged calling, dial-in bridging, multi-unicast to multicast transitions, ad-hoc bridge invitation and transition, and various forwarding variations [5].

As an example of these extension and service creation mechanisms, the PSTN and Internet Internetworking (pint) working group in IETF is defining a simple SIP extension for click-to-call type of services. In this scenario, a user at a web page clicks on a button, and a PSTN entity connects the user's telephone to a customer service rep. This requires a control protocol between the web server and a PSTN-enabled device. SIP is being used as this protocol.

H.323 does provide some basic mechanisms along this line. The FACILITY message allows a callee to direct a caller to contact a different party (basically, a blind transfer). Another is the H.245 CommunicationModeCommand, which allows the MC to change the media encodings for a conference for the various participants. The former is fairly limited in scope, and the latter can only be executed by the MC for the call. Neither provide generic third party control mechanisms needed for building complex services.

Another aspect of extensibility is modularity. Internet telephony requires a large number of different functions; these include basic signaling, conference control, quality of service, directory access, service discovery, etc. One can be certain that mechanisms for accomplishing these functions will evolve over time (especially with regards to QoS). This makes it critical to apportion these functions to seperate, modular, orthogonal components, which can be swapped in and out over time. It is also critical to use seperate, general protocols for each of these functions. This allows for the function to be duplicated in other applications with ease. For example, it is more efficient to have a single QoS mechanism which is application independent, rather than invent a new QoS protocol or mechanism for each application.

SIP is reasonably modular. It encompasses basic call signaling, user location, and registration. Advanced signaling is part of SIP, but within a single extension. Quality of service, directory accesses, service discovery, session content description, and conference control are all orthogonal, and reside in separate protocols. For example, it is possible to use the H.245 capability description elements in SIP, with no changes to SIP at all.

H.323 is less modular. It defines a vertically integrated protocol suite for a single application. The mix of services provided by the H.323 components encompass capability exchange, conference control, maintenance operations, basic signaling, quality of service, registration, and service discovery. Furthermore, these are intertwined within the various sub-protocols within H.323.

SIP's modularity allows it to be used in conjunction with H.323. A user can use SIP to locate another user, taking advantage of its rich multi-hop search facilities. When the user is finally located, they can use a redirect response to an H.323 URL, indicating that the actual communication should take place with H.323.

IV. Scalability

We also find that H.323 and SIP differ in terms of scalability. We can observe scalability on a number of different levels:

Large Numbers of Domains: H.323 was originally conceived for use on a single LAN. Issues such as wide area addressing and user location were not a concern. The newest version defines the concept of a zone, and defines procedures for user location across zones for email names. However, for large numbers of domains, and complex location operations, H.323 has scalability problems. It provides no easy way to perform loop detection in complex multi-domain searches (it can be done statefully by storing messages, which is not scalable). SIP, however, uses a loop detection algorithm similar to the one used in BGP, which can be performed in a stateless manner.

Server Processing: In an H.323 system, both telephony gateways and gatekeepers will be required to handle calls from a multitude of users. Similarly, SIP servers and gateways will need to handle many calls. For large, backbone IP telephony providers, the number of calls being handled by a large server can be significant.

In SIP, a transaction through several servers and gateways can be either stateful or stateless. In the stateless model, a server receives a call request, performs some operation, forwards the request, and completely forgets about it. SIP messages contain sufficient state to allow for the response to be forwarded correctly. Furthermore, SIP can be carried on either TCP or UDP. In the case of UDP, no connection state is required. This means that large, backbone servers can be based on UDP and operate in a stateless fashion, reducing signficantly the memory requirements and improving scalability.

H.323, on the other hand, requires gatekeepers (when they are in the call loop), to be stateful. They must keep call state for the entire duration of a call. Furthermore, the connections are TCP based, which means a gatekeeper must hold its TCP connections for the entire duration of a call. This can pose serious scalability problems for large gatekeepers.

Furthermore, a gateway or gatekeeper will need to process the signaling messages for each call. The simpler the signaling, the faster it can be processed, and the more calls a gateway or gatekeeper can support. As SIP is simpler to process than H.323, SIP should allow more calls per second to be handled on particular box than H.323. [1]

Conference Sizes: H.323 supports multiparty conferences with multicast data distribution. However, it requires a central control point (called an MC) for processing all signaling, for even the smallest conferences. This presents several difficulties. Firstly, should the user providing the MC functionality leave the conference, and exit their application, the entire conference terminates. In addition, since MC and gatekeeper functionality is optional, H.323 cannot support even three party conferences in some cases. We note that the MC is a bottleneck for larger conferences. To alleviate this, the latest version of H.323 has defined the concept of cascaded MC's, allowing for a very limited application layer multicast distribution tree of control messaging. This improves scaling somewhat, but for even larger conferences, the H.332 protocol defines additional procedures. This means that three distinct mechanisms exist to support conferences of different sizes. SIP, however, scales to all different conference sizes. There is no requirement for a central MC; conference coordination is fully distributed. This improves scalability and complexity. Furthermore, as it can use UDP as well as TCP, SIP supports native multicast signaling, allowing a single

[1] The authors are not aware of any study measuring the processing overhead of SIP and H.323, however.

Feature	SIP	H.323
Blind Transfer	Yes	Yes
Operator Assisted Transfer	Yes	No
Hold	Yes; through SDP	Not yet
Multicast Conferences	Yes	Yes
Multi-unicast Conferences	Yes	Yes
Bridged Conferences	Yes	Yes
Forward	Yes	Yes
Call Park	Yes	No
Directed Call Pickup	Yes	No

TABLE I
SIP AND H.323 CALL CONTROL FEATURE COMPARISON

protocol to scale from sessions with two to millions of members.

Feedback: H.245 defines procedures that allow receivers to control media encodings, transmission rates, and error recovery. This kind of feedback makes sense in point-to-point scenarios, but ceases to be functional in multipoint conferencing. SIP, instead, relies on RTCP for providing feedback on reception quality (and also for obtaining group membership lists). RTCP, like SIP, operates in a fully distributed fashion. The feedback it provides automatically scales from a two person point to point conference to huge broadcast style conferences with millions of participants.

V. SERVICES

H.323 and SIP offer roughly equivalent services. Some of the call control services are listed in Table 1.

As can be seen from the chart, SIP and H.323 support similar services. A comparison in these dimensions is somewhat difficult, as new services are always being added to both SIP and H.323. We expect that the above table will be different upon printing of this paper.

In addition to call control services, both SIP (when used with SDP) and H.323, provide capabilities exchange services. In this regard, H.323 provides a much richer set of functionality. Terminals can express their ability to perform various encodings and decodings based on parameters of the codec, and based on which other codecs are in use. However, most implementations don't require (or implement) these, and the basic receiver-capability indication supported by SIP ("choose any subset of these encodings for this list of media streams") seems sufficient and equivalent to current H.323 capabilities actually implemented.

SIP provides rich support for personal mobility services, however. When a caller contacts the callee, the callee can redirect the caller to a number of different locations. Each of these locations can be an arbitrary URL, and contains additional information about the terminal at that location. Information on language spoken, business or home, mobile phone or fixed, and a list of callee priorities, can be conveyed for each location. This allows the caller flexibility in choosing which location to talk to. For non-interactive terminals, the original call setup can convey caller preferences about the nature of the terminal to be contacted. This allows network proxies to forward the call based on these preferences.

SIP also supports multi-hop "searches" for a user. When a call request is made to some particular address, a SIP server is contacted at that address. As this SIP server may not be the machine that the callee is currently residing at, the server can proxy the request to one or more additional servers. These servers, in turn, may further proxy the request until the party is contacted. A server can actually proxy the request to multiple servers in parallel. This allows the search for the user to operate more rapidly. SIP also allows multiple branches of the search to accept the call, passing the responses back to the caller. The caller can then decide which party to speak to. This would allow a call for `j.doe@company.com` to be picked up by both Mr. Doe, his wife, and an answering machine. The caller can then hang up with the answering machine and continue with a three party call, if they so desire.

H.323's support for this kind of mobility is more limited. The facility message can redirect a caller to try several other addresses (much like 300 class response codes in SIP). However, it cannot be used to express preferences, nor can the caller express preferences in the original call invitation. H.323 wasn't engineered for wide area operation; it does support forwarding of call requests among servers, but has no mechanisms for loop detection. H.323 doesn't allow a gatekeeper to proxy a request to multiple servers either.

H.323 supports various conference control services, including chair selection, "mike passing", and conference participant determination. SIP does not provide conference control, relying instead on other protocols for this service. Some simple forms of conference control (such as sending notes around, and obtaining a conference participant listing), are available through RTCP, however.

VI. CONCLUSION

In this paper, we have compared SIP and H.323 in terms of complexity, extensibility, scalability, and services. We have found that SIP provides a similar set of services to H.323, but provides far lower complexity, rich extensibility, and better scalability. Future work is to more fully evaluate the protocols, and examine quantitative performance metrics to characterize these differences.

REFERENCES

[1] M. Handley, H. Schulzrinne, and E. Schooler, "SIP: session initiation protocol," Internet Draft, Internet Engineering Task Force, May 1998, Work in progress.

[2] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," Request for Comments (Proposed Standard) 2068, Internet Engineering Task Force, Jan. 1997.

[3] H. Schulzrinne, R. Lanphier, and A. Rao, "Real time streaming protocol (RTSP)," Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force, Apr. 1998.

[4] Anonymous, "H.323 and firewalls: The problems and pitfalls of getting H.323 safely through firewalls," Developer note, Intel Corporation, Apr. 1997.

[5] Henning Schulzrinne and Jonathan Rosenberg, "Signaling for internet telephony," Technical Report CUCS-005-98, Columbia University, New York, New York, Feb. 1998.

Understanding SIP

Today's Hottest Communications Protocol
Comes of Age

WHITE PAPER

Ubiquity

Understanding SIP

Today's Hottest Communications Protocol Comes of Age

Introduction

The growing thirst among communications providers, their partners and subscribers for a new generation of IP-based services is now being quenched by SIP – the Session Initiation Protocol. An idea born in a computer science laboratory less than a decade ago, SIP is the first protocol to enable multi-user sessions regardless of media content and is now a specification of the International Engineering Task Force (IETF).

Today, increasing numbers of carriers, CLECs and ITSPs are offering such SIP-based services as local and long distance telephony, presence & Instant Messaging, IP Centrex/Hosted PBX, voice messaging, push-to-talk, rich media conferencing, and more. Independent software vendors (ISVs) are creating new tools for developers to build SIP-based applications as well as SIP software for carriers' networks. Network equipment vendors (NEVs) are developing hardware that supports SIP signaling and services. There is a wide variety of IP phones, User Agents, network proxy servers, VOIP gateways, media servers and application servers that all utilize SIP.

Gradually, SIP is evolving from the prestigious protocols it resembles -- the Web's Hyper Text Transfer Protocol (HTTP) formatting protocol and the Simple Mail Transfer Protocol (SMTP) email protocol -- into a powerful emerging standard. However, while SIP utilizes its own unique user agents and servers, it does not operate in a vacuum. Comparable to the converging of the multimedia services it supports, SIP works with a myriad of preexisting protocols governing authentication, location, voice quality, etc.

This paper provides a high-level overview of what SIP is and does. It charts SIP's migration from the laboratory to the marketplace. It describes the services SIP provides and the initiatives underway that will spur its growth. It also details the key features that distinguish SIP among protocols and diagrams how a SIP session takes place.

A New Generation of Services

Flexible, extensible and open, SIP is galvanizing the power of the Internet and fixed and mobile IP networks to create a new generation of services. Able to complete networked messages from multiple PCs and phones, SIP establishes sessions much like the Internet from which it was modeled.

In contrast to the longstanding International Telephony Union (ITU) SS7 standard used for call setup and management and the ITU H.323 video protocol suite, SIP operates independent of the underlying network transport protocol and is indifferent to media. Instead, it defines how one or more participant's end devices can create, modify and terminate a connection whether the content is voice, video, data or Web-based.

SIP is a major upgrade over protocols such as the Media Gateway Control Protocol (MGCP), which converts PTSN audio signals to IP data packets. Because MGCP is a closed, voice-only standard, enhancing it with signaling capabilities is complex and at times has resulted in corrupted or discarded messages that handicap providers from adding new services. Using SIP, however, programmers can add new bits of information to messages without compromising connections.

For example, a SIP service provider could establish an entirely new medium consisting of voice, video and chat. With MGCP, H.323 or SS7, the provider would have to wait for a new iteration of the protocol to support the new medium. Using SIP, a company with locations on two continents could enable the medium, even though the gateways and devices may not recognize it.

Moreover, because SIP is analogous to HTTP in the way it constructs messages, developers can more easily and quickly create applications using popular programming languages such as Java. Carriers who waited years to deploy call-waiting, caller ID and other services using SS7 and the Advanced Intelligent Network (AIN) can deploy premium communications services in just months with SIP.

This level of extensibility is already making its mark in growing numbers of SIP-based services. Vonage, a service provider targeting consumer and small business customers, delivers over 20,000 lines of digital local and long distance calling and voice mail to over customers using SIP. Deltathree, which provides Internet telephony products, services and infrastructure for service providers, offers a SIP-based PC-to-Phone solution that lets PC users call any phone in the world. Denwa Communications, which wholesales voice services worldwide, delivers PC to PC and Phone to PC caller ID, voice mail as well as conference calling, unified messaging, account management, self-provisioning and Web-based personalized services using SIP.

Understanding SIP

Today's Hottest Communications Protocol Comes of Age

While some pundits predict that SIP will be to IP what SMTP and HTTP are to the Internet, others say it could signal the end of the AIN. To date, the 3G Community has selected SIP as the session control mechanism for the next-generation cellular network. Microsoft has chosen SIP for its real-time communications strategy and has deployed it in Microsoft XP, Pocket PC and MSN Messenger. Microsoft also announced that its next version of CE.net will include a SIP-based VoIP application interface layer, and is committed to deliver SIP-based voice and video calls to consumers' PCs.

In addition, MCI is using SIP to deploy advanced telephony services to its IP communications customers. Users will be able to inform callers of their availability and preferred method of communication, such as email, telephone or Instant Message. Presence will also enable users to instantly set up chat sessions and audio-conferences. With SIP, the possibilities go on and on.

A Historical Snapshot

SIP emerged in the mid-1990s from the research of Henning Schulzrinne, Associate Professor of the Department of Computer Science at Columbia University, and his research team. A co-author of the Real-Time Transport Protocol (RTP) for transmitting real-time data via the Internet, Professor Schulzrinne also co-wrote the Real Time Streaming Protocol (RTSP) -- a proposed standard for controlling streaming audio-visual content over the Web.

Schulzrinne's intent was to define a standard for Multi-party Multimedia Session Control (MMUSIC). In 1996, he submitted a draft to the IETF that contained the key elements of SIP. In 1999, Shulzrinne removed extraneous components regarding media content in a new submission, and the IETF issued the first SIP specification, RFC 2543. While some vendors expressed concerned that protocols such as H.323 and MGCP could jeopardize their investments in SIP services, the IETF continued its work and issued SIP specification RFC 3261 in 2001.

The advent of RFC 3261 signaled that the fundamentals of SIP were in place. Since then, enhancements to security and authentication among other areas have been issued in several additional RFCs. RFC 3262, for example, governs Reliability of Provisional Responses. RFC 3263 establishes rules to locate SIP Proxy Servers. RFC 3264 provides an offer/answer model and RFC 3265 determines specific event notification.

As early as 2001, vendors began to launch SIP-based services. Today, the enthusiasm for the protocol is growing. Organizations such as Sun Microsystems' Java Community Process are defining application program interfaces (APIs) using the popular Java programming language so developers can build SIP components and applications for service providers and enterprises. Most importantly, increasing numbers of players are entering the SIP marketplace with promising new services, and SIP is on path to become one of the most significant protocols since HTTP and SMTP.

The SIP Advantage: Open, Extensible Web-Like Communications

Like the Internet, SIP is easy to understand, extend and implement. As an IETF specification, SIP extends the open-standards spirit of the Internet to messaging, enabling disparate computers, phones, televisions and software to communicate. As noted, a SIP message is very similar to HTTP (RFC 2068). Much of the syntax in message headers and many HTTP codes are re-used. Using SIP, for example, the error code for an address not found, "404," is identical to the Web's. SIP also re-uses the SMTP for address schemes. A SIP address, such as sip:guest@sipcenter.com, has the exact structure as an email address. SIP even leverages Web architectures, such as Domain Name System or Service (DNS), making messaging among SIP users even more extensible.

Using SIP, service providers can freely choose among standards-based components and quickly harness new technologies. Users can locate and contact one another regardless of media content and numbers of participants. SIP negotiates sessions so that all participants can agree on and modify session features. It can even add, drop or transfer users.

However, SIP is not a cure-all. It is neither a session description protocol, nor does it provide conference control. To describe the payload of message content and characteristics, SIP uses the Internet's Session Description Protocol (SDP) to describe the characteristics of the end devices. SIP also does not itself provide Quality of Service (QoS) and interoperates with the Resource Reservation

Setup Protocol (RSVP) for voice quality. It also works with a number of other protocols, including the Lightweight Directory Access Protocol (LDAP) for location, the Remote Authentication Dial-In User Service (RADIUS) for authentication and RTP for real-time transmissions, among many others.

SIP provides for the following basic requirements in communications:

1. User location services
2. Session establishment
3. Session participant management
4. Limited feature establishment

An important feature of SIP is that it does not define the type of session that is being established, only how it should be managed. This flexibility means that SIP can be used for an enormous number of applications and services, including interactive gaming, music and video on demand as well as voice, video and Web conferencing.

Below is are some of other SIP features that distinguish it among new signaling protocols

- SIP messages are text based and hence are easy to read and debug. Programming new services is easier and more intuitive for designers.
- SIP re-uses MIME type description in the same way that email clients do, so applications associated with sessions can be launched automatically.
- SIP re-uses several existing and mature internet services and protocols such as DNS, RTP, RSVP etc. No new services have to be introduced to support the SIP infrastructure, as much of it is already in place or available off the shelf.
- SIP extensions are easily defined, enabling service providers to add them for new applications without damaging their networks. Older SIP-based equipment in the network will not impede newer SIP-based services. For example, an older SIP implementation that does not support method/ header utilized by a newer SIP application would simply ignore it.
- SIP is transport layer independent. Therefore, the underlying transport could be IP over ATM. SIP uses the User Datagram Protocol, (UDP) as well as the Transmission Control Protocol (TCP) protocol, flexibly connecting users independent of the underlying infrastructure.
- SIP supports multi-device feature levelling and negotiation. If a service or session initiates video and voice, voice can still be transmitted to non-video enabled devices, or other device features can be used such as one way video streaming.

The Anatomy of a SIP Session

SIP sessions utilize up to four major components: SIP User Agents, SIP Registrar Servers, SIP Proxy Servers and SIP Redirect Servers. Together, these systems deliver messages embedded with the SDP protocol defining their content and characteristics to complete a SIP session. Below is a high-level description of each SIP component and the role it plays in this process.

SIP User Agents (UAs) are the end-user devices, such as cell phones, multimedia handsets, PCs, PDAs, etc. used to create and manage a SIP session. The User Agent Client initiates the message. The User Agent Server responds to it.

SIP Registrar Servers are databases that contain the location of all User Agents within a domain. In SIP messaging, these servers retrieve and send participants' IP addresses and other pertinent information to the SIP Proxy Server.

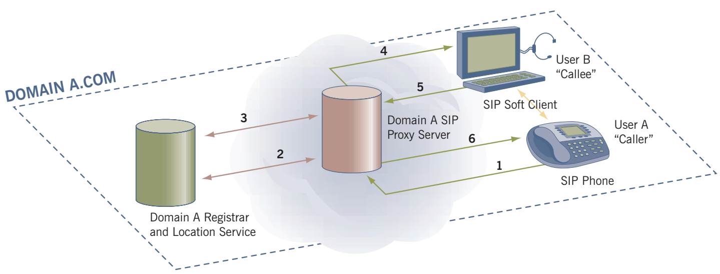
SIP Proxy Servers accept session requests made by a SIP UA and query the SIP Registrar Server to obtain the recipient UA's addressing information. It then forwards the session invitation directly to the recipient UA if it is located in the same domain or to a Proxy Server if the UA resides in another domain.

SIP Redirect Servers allow SIP Proxy Servers to direct SIP session invitations to external domains. SIP Redirect Servers may reside in the same hardware as SIP Registrar Severs and SIP Proxy Servers.

The following scenarios demonstrate how SIP components work in harmony to establish SIP sessions between UAs in the same and different domains:

Establishing A SIP Session Within the Same Domain
The diagram below illustrates the establishment of a SIP

session between two users who subscribe to the same ISP and, hence, use the same domain. User A relies on a SIP phone. User B has a PC running a soft client that can support voice and video. Upon powering up, both users register their availability and their IP addresses with the SIP Proxy Server in the ISP's network. User A, who is initiating this call, tells the SIP Proxy Server he/she wants to contact User B. The SIP Proxy Server then asks for and receives User B's IP address from the SIP Registrar Server. The SIP Proxy Server relays User A's invitation to communicate with User B, including -- using SDP -- the medium or media User A wants to use. User B informs the SIP Proxy Server that User A's invitation is acceptable and that he/she is ready to receive the message. The SIP Proxy Server communicates this to User A, establishing the SIP session. The users then create a point-to-point RTP connection enabling them to interact.



1. Call User B
2. Query "Where is User B?"
3. Response "User B SIP Address"
4. 'Proxied' Call
5. Response
6. Response
7. Multimedia Chanel Establised

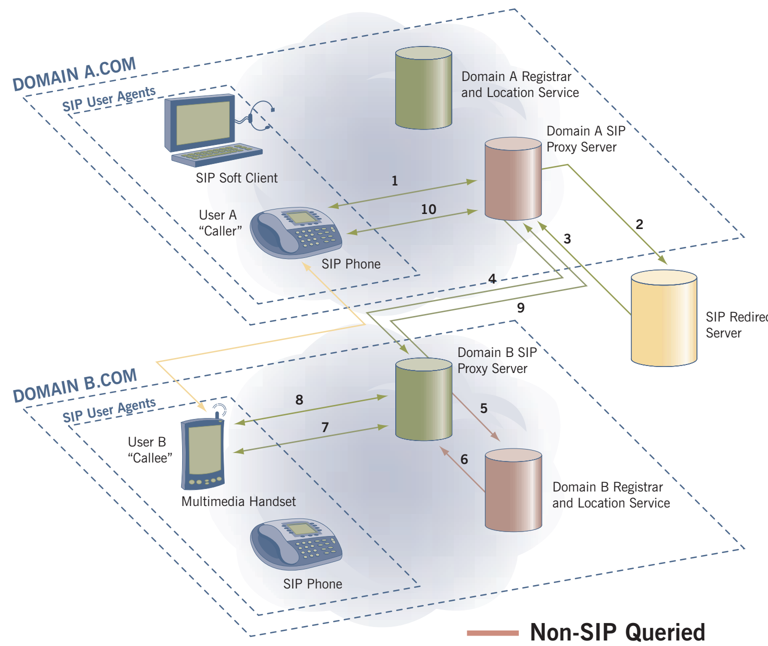
Non-SIP Queried (i.e. Database Lookup)

SIP Signaling

RTP

Establishing A SIP Session In Dissimilar Domains

The difference between this scenario and the first is that when User A invites User B -- who is now using a multimedia handset -- for a SIP session the SIP Proxy Server in Domain A recognizes that User B is outside its domain. The SIP Proxy Server then queries the SIP Redirect Server -- which can reside in either or both Domain A or B -- for User B's IP address. The SIP Redirect Server feeds User B's contact information back to the SIP Proxy Server, which forwards the SIP session invitation to the SIP Proxy Server in Domain B. The Domain B SIP Proxy Server delivers User A's invitation to User B, who forwards his/her acceptance along the same path the invitation travelled.



Non-SIP Queried (i.e. Database Lookup)

SIP Signaling

RTP

1. Call User B
2. Query "How do I get to User B, Domain B?"
3. Response "Address of Proxy Controller for Domain"
4. Call 'Proxied' to SIP Proxy for Domain B
5. Query "Where is User B?"
6. User B's Address
7. Proxied Call
8. Response
9. Response
10. Response
11. Multimedia Channel Established

Seamless, Flexible, Extensible: Looking Ahead With SIP

Able to connect users across any IP network (wireline LAN and WAN, the public Internet backbone, mobile 2.5G, 3G and Wi-Fi and any IP device (phones, PCs, PDAs, mobile handsets), SIP opens the door to a wealth of lucrative new possibilities that improve how businesses and consumers communicate. Used alone, SIP-based applications such as VOIP, rich media conferencing, push-to-talk, location-based services, Presence and IM offer service providers, ISVs, network equipment vendors and developers a plethora of new commercial opportunities. However, SIP's ultimate value lies in its ability to combine these capabilities as subsets of larger, seamless communications services.

Using SIP, service providers and their partners can customize and deliver a portfolio of SIP-based services that let subscribers use conferencing, Web controls, Presence, IM and more within a single communications session. Service providers can, in effect, create one flexible application suite that addresses many end user needs instead of installing and supporting discrete, "stovepipe" applications that are tied to narrow, specific functions or types of end devices.

By consolidating their IP-based communications services under a single, open standards-based SIP application framework, service providers can dramatically lower the cost of designing and deploying innovative new IP-based hosted services to their customers. This is the power SIP's extensibility can bring to the industry and the marketplace and the promise it holds out for us all.

cnet NEWS.COM http://www.news.com/

Cisco adopts IP telephony standard

By Marguerite Reardon
http://news.com.com/Cisco+adopts+IP+telephony+standard/2100-7352_3-6046275.html

Story last modified Mon Mar 06 10:36:53 PST 2006

Cisco Systems plans to finally adopt a key Internet Protocol telephony standard, allowing the addition of new network-service features and enabling companies such as Microsoft to integrate their communications products with Cisco gear.

On Monday, at the VoiceCon 2006 conference in Orlando, Fla., Cisco said it will add support for session initiation protocol, or SIP, to its IP PBX software. The new version of the product, CallManager 5.0, will include SIP capabilities for Cisco IP phones, presence-awareness software and multimedia communications software.

SIP is used to establish contact between IP phones and to add special features--such as presence awareness, video or mobility capabilities--onto a voice over Internet Protocol (VoIP) network. The standard also makes it possible for companies deploying VoIP to mix and match the products they use, significantly lowering the cost of deploying a VoIP network.

Cisco had been the only major supplier in the market not to support SIP in its IP PBX software. Cisco sees the addition of SIP as an important step in being able to provide customers more features.

"IP telephony isn't just about toll bypass anymore," said Barry O'Sullivan, vice president of IP communications for Cisco. "It's about improving productivity and allowing people to do their jobs more effectively. And people need to be able to communicate and collaborate through the means that suits them best."

CallManager 5.0 should work with any SIP-based phone, but Cisco said specifically it plans to support a "softphone" (or PC-based phone) client for Research In Motion's BlackBerry handheld as well as Nokia's new dual-mode phones.

In addition to the upgraded CallManager, Cisco announced other new products including the Unified Presence Server, which collects status and availability data from users' devices and feeds it to Cisco applications, and the Unified Personal Communicator, which allows users to see on their PCs or IP phones who is online.

As part of the announcement this week, Cisco said it is working with Microsoft to integrate its Office Communicator 2005 and Office Live Communications with Cisco's Unified Communications System. The integration means that users can launch a VoIP conversation directly from their Microsoft Outlook client. The interoperable package should be available in August 2006, the companies said.

http://www.nortel.com/corporate/news/newsreleases/2004b/06_22_04_open_client_strategy.html

June 22, 2004
Nortel Networks Sets Stage for Mass Deployment of Multimedia Communications with Open Client Strategy
i3 micro, Polycom, Texas Instruments, Uniden Announce Support for Global SIP Initiative

CHICAGO – Nortel Networks* [NYSE/TSX: NT], building on its Session Initiation Protocol (SIP) leadership, announced a global initiative designed to dramatically increase the market adoption and deployment of SIP-based multimedia services in consumer and enterprise markets.

As part of its open client strategy, Nortel Networks will make available a comprehensive documentation suite to enable third-party development and testing of SIP-based clients and terminals interoperable with Nortel Networks Multimedia Communication Server (MCS) 5100 and 5200 products.

In support of this initiative to promote industry-wide SIP adoption, Texas Instruments Incorporated [NYSE:TXN] plans to adapt its voice over Internet Protocol (VoIP) reference designs for SIP-based customer premises equipment (CPE) such as terminal adapters, VoIP gateways and IP phones to achieve Nortel Networks MCS interoperability. In addition, third-party vendors Uniden America Corporation, i3 micro, and Polycom* [NASDAQ: PLCM] are developing or plan to develop MCS interoperable SIP clients and terminals in their consumer and enterprise products.

SIP is strategic to mass market deployment of multimedia services because it brings Internet-style innovation to the traditional world of carrier voice services. SIP establishes real-time communication sessions in IP networks that contain any combination of media (voice, video, data, etc.) and can be as basic as a voice call or as complex as a multi-party mixed media conference. With SIP, service providers can harness the power and ubiquity of IP to create innovative new services that combine elements from telephony and other Web applications like e-mail, messaging, presence and video streaming.

“Today’s announcement further supports Nortel Networks global vision of transforming networks, eliminating boundaries and enriching communications,” said Sue Spradley, president, Wireline Networks, Nortel Networks. “Making our SIP client interoperability specification available to a broad range of client and terminal vendors across the industry helps accelerate the delivery of multimedia communications to mainstream consumers and enterprises, putting SIP clients in every home and on every desktop.”

As a leading provider of software and digital signal processing (DSP) technology for VoIP SIP CPE devices, Texas Instruments will offer MCS interoperable reference designs to a vast array of IP phone and CPE manufacturers and designers that use its

VoIP solutions. This initiative will significantly expand the availability of MCS interoperable clients to all markets – service providers, enterprise and consumers.

Uniden, a leading manufacturer of wireless consumer electronics products, also intends to make its suite of enterprise SIP terminals and a planned suite of SIP consumer VoIP solutions interoperable with Nortel Networks MCS client.

i3 micro technology, a leading provider of voice over IP (VoIP) CPE products and related management solutions, plans to extend its existing Vood* (Voice Options on Demand) VoIP product interoperability with Nortel Networks to include the MCS SIP specification. Nortel Networks open client strategy and i3 micro's CPE-based value added SIP applications will drive new revenue opportunities for service providers and their MCS customers.

Polycom, a leader in unified collaborative communications solutions, is implementing MCS SIP compatibility on its line of Polycom VSX video conferencing systems, its MGC voice and video conference bridges and on its series of SoundPoint* IP desktop phones.

Service providers will benefit from Nortel Networks open client strategy by being able to more effectively address the divergent needs of the consumer, small, medium and large enterprise marketplaces with a wider range of CPE choices for their customers and more flexible packaging options.

Bell Canada, which recently launched its Managed IP Telephony solution, is leading the way as businesses move to an IP world. Bell Canada will be positioned to offer customers a greater variety of SIP-based communications devices. In addition, through creation of joint innovation centers with Nortel Networks last year, Bell Canada and Nortel Networks continue to focus on creating new services based on IP telephony and multimedia capabilities.

As part of its open client strategy, Nortel Networks also plans to provide vendors with a Theme Designer Kit (TDK) that will allow vendor participants to modify the look and feel of the MCS multimedia PC soft client. The TDK tool will make it possible to easily modify the format of the client to meet service provider branding requirements or provide end users with the ability to customize the look and feel of the client by selecting different combinations of colors, backgrounds and other thematic elements. The TDK is expected to be available in the fourth quarter of 2004.

Vendors wanting to participate in Nortel Networks industry-wide SIP initiative can request access to the MCS documentation suite by completing an application available on Nortel Networks Developer Program Web site. In the future, vendors will also be able to access the TDK from the same location.

Nortel Networks is a worldwide leader in delivering SIP innovation with customers and partners like Bell Canada, Charter Communications, Dacom, Erlanger Health System, the

FedEx Institute of Technology at the University of Memphis, the University of Michigan, Monster, OneConnect, SBC, SaskTel, Sungard, TeliaSonera, Texas A&M University and Verizon Communications.

Nortel Networks Multimedia Communications Portfolio, including both MCS 5100 and MCS 5200, delivers advanced multimedia and collaborative applications through the same commercially available hardware and open-standards software. This portfolio delivers the scale and functionality necessary for both enterprises and service providers to address their target markets. It transforms the way users communicate by enabling next generation tools that improve productivity and facilitate ubiquitous access to communications services. Nortel Networks will be demonstrating its MCS 5200 product on booth #11326 during SUPERCOMM 2004 in Chicago.

For the entire year of 2003 and the first quarter of 2004, Nortel Networks ranked #1 in the global markets for voice over packet ports shipped and global softswitch revenue, according to Synergy Research Group.

Nortel Networks has a proven portfolio of products and services for packet voice and multimedia services. Nortel Networks is providing Nortel Networks Succession* voice over packet solutions to a number of leading operators, including Bell Canada, Cable & Wireless Cayman Islands, Charter Communications, China Netcom, China Railcom, Cox Communications, Hong Kong Broadband Network, MCI, Sprint and Verizon Communications.

Nortel Networks is an industry leader and innovator focused on transforming how the world communicates and exchanges information. The Company is supplying its service provider and enterprise customers with communications technology and infrastructure to enable value-added IP data, voice and multimedia services spanning Wireless Networks, Wireline Networks, Enterprise Networks, and Optical Networks. As a global company, Nortel Networks does business in more than 150 countries. More information about Nortel Networks can be found on the Web at www.nortelnetworks.com or www.nortelnetworks.com/media_center.

Issue :: May 2005

SIP Supreme!

No doubt about it. SIP is the greatest technological success story in the IP Communications industry. by Richard Grigonis

SIP is well on its way to becoming the single most dominant (not to mention recognizable) protocol in the VoIP world. This call control protocol more than overshadows its predecessor and former rival, H.323.

Why SIP? The reasons for this are quite simple, as revealed by the following anecdote told by Joan Vandermat, VP of product management at Siemens: "About five years ago, we made huge efforts to incorporate H.323 into our old HiPath 5000 and the HiNet RC2000, an NT serverbased call processor that did direct signaling to LANconnected H.323 peripherals, such as the LP5100 or the Siemens ComDesk PC client soft-phone option. In those days we were evangelistic about being 100 percent H.323 compliant in the client. But it was a bear to develop on. It took too long and too much code to bring new features into the platforms."

"About four years ago, we began to investigate the SIP trail," says Vandermat. "We tasked a few of our engineers who had worked on the H.323 products to see what they could do with SIP. We gave them a pretty low threshold. We just wanted to see if they could take our H.323 phones and program them to do some basic call setup and teardown using SIP. In just two days, they had not only basic call control, but many features were operational on the phones too. We were awestruck."

"There weren't too many SIP products on the market in those days," says Vandermat. "but we found a few SIP products, brought them into the lab and said 'let's see if we can make these work with our equipment.' H.323 had notorious interoperability problems, and we wondered how SIP would fare. Amazingly, two out of the three off-the-shelf products we found worked instantly with our SIP-enabled equipment. Our engineers, who knew H.323 intimately, said SIP was far easier to work with than H.323. More important, SIP was also far more efficient than H.323. The performance of our first generation IP phones under H.323 was not terrific, but when we ran SIP on them, call setup and teardown times were cut in half. We were amazed with SIP and were sold on it immediately."

Today, Siemens has a lot of confidence in SIP's future.

"We hope and expect SIP will become the dominant protocol, to almost the exclusion of other protocols," says Vandermat. "In fact, we're embedding SIP call control into every part of our portfolio. We have a relationship with Microsoft and their Live Communications Server, we're already shipping OpenScape our flagship application that has SIP at its heart, and we offer the HiPath 8000 softswitch, which in the past has been offered on the carrier side and is now being offered in a scaled-down, SIP-based version for the enterprise market."

Vandermart continues: "We're adding an internal SIP gateway to our HiPath 3000 and 4000 platforms, which are more 'hybird' or 'converged' platforms, and which are commonly called IP PBXs. Those are currently H.323 architected with lots of proprietary extensions, but we are now adding an internal SIP media gateway or 'soft gateway' inside of both the 3000 and 4000. It won't be a bolt-on external server, it will be a piece of integral software, so that those platforms will work in SIP environments, either on the station side (you'll be able to integrate SIP stations such as Windows PCs or SIP phones from some third party vendor) and on the trunk/network side, so a business could buy SIP-based trunking from a carrier. The HiPath 4000 will be able to do this later this year and the HiPath 3000 will get the capability in early 2006."

"We're even going back and SIP-enabling our ProCenter contact center suite, and we'll bring forward our Xpressions unified messaging so it can participate in a SIP environment too," says Vandermart. "You can use Xpressions either with our own HiPath SIP softswitch or you could use somebody else's SIP-enabled softswitch."

"Within about 15 months, every single product in the Siemens portfolio will either be running SIP natively or will be SIP-enabled," says Vandermart. "I don't think any other vendor is being that aggressive in terms of putting R&D money behind SIP as well as actual marketing deployment and support. Siemens has bet the farm on SIP and there's other big money out there behind SIP too."

You Can Get It Wholesale

Indeed, SIP has taken on a sort of commodity/utility status with the announcement by SimpleTelecom (www.simpletelecom.com) of wholesale SIP services now available through a web-portal (www.simpletelecom.com/simplecarrier) at competitive prices. SimpleCarrier, officially launched in March 2005, entered the wholesale IP Voice market as the first carrier-class wholesale SIP service with instant activation and no minimum usage commitment. Using a web-based Dashboard (for reporting, configuration and technical support), carriers and major IP Voice users can, in about five minutes, create an account and start using their SIP termination service.

Beyond Voice

Just as IP Communications is more than VoIP, SIP isn't just voice either. It supports services that provide interactive multimedia-based personal communications environments of which voice is but a component.

One example of this is CPT International (www.cptii.com), which hosts VoiceXML and SIP-based services. Voice Harbor is CPT's new, standards based telephony platform, providing hosting capacity for VoiceXML, speech, multi-modal and traditional touchtone-only telephony applications. Voice Harbor supports ASR (Automated Speech Recognition), text-to-speech synthesis, and speaker verification.

Mark Rayburn, director of advanced technology at CPT, says "We have a different perspective on SIP. The way we've used SIP so far is within our infrastructure. One reason we use SIP is to cut costs. We've been in the carrier space for over 12 years with proprietary products like many others. But when VoiceXML, the 'IVR of the future' came out, the fact that it was based on web technologies meant that for the first time you could separate the application from the telephony infrastructure. You could use the same guys who built your web applications; they just had to learn the VoiceXML tags. So if they're running a system on JBoss [A popular open source Java application server that supports the J2EE specifi- cations] or the BEA Weblogic server, and they're hitting the backend databases, the logic will be familiar."

"This time, however, instead of producing HTML of XML pages that go out to a browser, they can produce VoiceXML pages that go out to a VoiceXML gateway," says Rayburn. "Our company houses VoiceXML gateways and the speech servers that do the speech recognition, text-tospeech prompts, the switching and the transport. Basically, we take all of the 'headache stuff' that they used to have to deal with and push it out to a datacenter. Also, carriers can get out of the CapEx problem, and basically 'pay by the drink' with OpEx money, and yet they still keep full control of the application. We get the input from the caller, send it to the application server and go back and forth until the job is done."

"As for SIP, when a call comes in and it's a touchtone application, you have to convert tones into some text so you can use the digits in your business logic," says Rayburn. "That used to require a DSP board from Dialogic, NMS Communications or Brooktrout. If we go SIP-based, however, then the digits come in as an RTP [Real-Time Protocol] payload in accordance with RFC 2833, so we can eliminate that board from our VoiceXML gateways. That's a cost savings, first of all because we don't need a board, and secondly because we can now buy servers that don't need a PCI slot for the DSP card, and we don't have hardware integration issues involving the board. So SIP gives us a lot of cost savings."

"SIP headers enable me to transfer all kinds of information about a call around our system, and it's really allowed us to do many different flexible things and cut costs at the same time. I often joke that, for us, SIP is like a new kind of duct tape-you can use it in many different ways."

"As for taking SIP out to the network, the what's driving us in that direction is local number presence. Since we're running a hosted environment, there are many vertical applications that require a local number. Small banks, retail or government agencies want to keep their local number, rather than use an 800 number. To help them, I can go through a Level 3 or someone like that, who has the local number, and I can backhaul that voice via VoIP to a centralized hosting facility. It's key for us to bring local number presence to applications and verticals that used to be restricted to a customer premise."

Proprietary Extensions

To take SIP where no protocol has gone before generally requires making proprietary extensions to the code.

Stalker Software (www.stalker.com) makes SIP-based messaging and collaboration solutions for a various operating systems. The can meet the needs of any size operation, from a small office of 25 collaboration and email users to Tier 1 service providers hosting millions of accounts. Their flagship product is CommuniGate Pro, a scaleable, enterprise-capable messaging server.

Vladimir Butenko, Stalker's president and CEO, says that "some companies feel that, in order to distinguish their products from competitors, they must add unique features that necessitate adding proprietary extensions to the SIP standard. The only real player who's trying to really push proprietary extensions is Microsoft, as usual. But I don't think that's a very big problem. In my opinion, Microsoft developed its extensions not because it wanted to differentiate itself, but because there was a lack of standard code for various functions. Indeed, this void has still not been filled. SIP still does not cover certain essential issues such as security, though you're beginning to see such things as SIPS [SIP-Secure). On the presence side there are some useful technologies appearing, such as SIMPLE [SIP for Instant Messaging and Presence Leveraging Extensions]. Microsoft probably just didn't want to wait, so they moved forward with their own SIP extensions. They haven't documented everything, but I will tell you that we support their extensions. It's not really rocket science or a trade secret."

"I don't believe that providing proprietary extensions is a real way for companies to differentiate themselves," says Butenko. "At this stage, SIP is mature enough so that interoperability becomes a key issue. Nobody wants to buy a system that is too proprietary. There are many opportunities for vendors to differentiate themselves without resorting to going outside the SIP standard. I agree that certain problems still exist in terms of Presence and other things, but these will eventually be worked out in less than a year. Those vendors daring to take the proprietary route will be suffering just a few years down the road. Companies that introduced some proprietary technology and tried to use it as a key selling point of their product have realized their error and are already moving to provide the same features within the SIP standard."

"The SIP protocol and the SIMPLE presence technology are very complex but they also very flexible. Many things can be achieved without going outside the scope of the protocol. But there are still problems, of course. The main SIP problem still seems to be getting IP calls through NAT (Network Address Translation) and net traversal. So some companies build proxies and you see whole companies founded just to solve the very simple problem of net traversal. These vendors will charge you something outrageous like $30,000 for just a simple Linux box that they say is a proprietary solution; such boxes enable you to make only about 200 proxy connections through the net. This is funny. It's what happens in emerging markets."

"Our SIP proxy offering has all of this functionality built in," says Butenko. "We consider it a basic infrastructure feature. If you install it in the enterprise, either on the border or close to the net border, it automatically does all SIP traversal for the SIP protocol, all media proxies for voice, video and even TCP protocols. As a result, you can not only use Windows Messenger with our proxy product, but remote assistance, and all the other features that Messenger supports. If you install it on a large telco or ISP site, then it does so-called Far-End NAT traversal."

SIP's Secure Future?

"I'd like to see SIP eventually handle the authentication of the user agent," says CPT's Rayburn. "They might have to change the RFC 3261 'standard' or else 'wrap' some code around it. Security is definitely a huge issue; maybe SIPSecure will address it. Also, a lot of people want to lose the session border controllers for carrier-peering, and incorporate SBC functions directly into the protocol itself in some way, to boost security and eliminate yet another component they must buy. Something like that might warrant a whole new revision of SIP."

Richard Grigonis is Editor-in-Chief of VON Magazine.

SIP in the News...

In scanning through recent news items crossing our desk, the acronym SIP appears as often as does VoIP…

Hotsip (www.hotsip.com) is a SIP Application Server provider that offers large-scale SIP-enabled broadband and 3G/IMS (IP-based Multimedia Subsystem) networks. It recently launched an SCE (Service Creation Environment), the Hotsip® Multimedia Communication Engine (M2CET) SCE, loaded with open APIs, for the development of SIPbased applications running in fixed, mobile, broadband and converged networks.

Intertex Data AB (www.intertex.se) has developed what's said to be the world's first SIP aware broadband firewall and NAT-router. The IX66 Internet Gate includes a SIP proxy and SIP registrar dynamically controlling the firewall. Intertex's SIP Switch upgrade adds PBX functionality, voice, video, Presence, instant messaging and Microsoft Messenger are supported. An optional ADSL modem can be included.

Intoto (www.intoto.com) is a software ODM that provides converged security, VoIP and WLAN functionality to over 100 OEMs, including Netgear, which uses Intoto's flagship iGateway software platform to power its upcoming AT&T CallVantage router. Intoto's iGateway is an embedded software module that includes various voice protocols with SIP signaling and a generic interface to voice cards. iGateway VoIP architecture can be used to build IP phones, IADs, IP PBXs, voice gateways and media gateways for organizations various sizes.

IPeria (www.iperia.com) offers ActivEdge, a software package providing SIP-based scalable voicemail, an automated attendant, unified messaging and conferencing applications for PSTN and IP networks. The Visual Voicemail component allows subscribers to access voicemail messages with a computer and Internet connection as they would email. ActivEdge works with off-the-shelf hardware and is suited for network service, wireless, and cable providers.

Longboard (www.longboard.com) makes open mobile convergence software solutions such as their OME (Open Mobile Enterprise) server-to-handset solutions enabling wireline carriers to deliver IP Fixed Mobile Convergence services seamlessly across WiFi and cellular networks. Longboard's LMAP platform provides a SIP-based Applications Server that can include a service creation capability allowing pre-packaged features to be customized and extended.

MediaRing (www.mediaring.com) sells the VoizBridge Session Border Contoller, which offers VoIP protocol interworking (H.323 and SIP).

MediaTrix (www.mediatrix.com) is known for their residential IP gateways but they also have products used by SMBs and enterprise remote offices. They also offer a SIP server.

net.com (www.net.com) offers the SHOUT platform, which connects TDM PBXs over IP networks. It has both SIP and H.323 support.

Net2Phone (www.net2phone.com) provides PacketCablecompliant SIP and wireless VoIP solutions worldwide.

Pactolus Communications Software (www.pactolus.com) provides IP voice services to carriers for deployment in SIP-enabled converged and end-to-end IP networks. Their RapidFLEX product includes a SIP-based Application Server, Service Creation Environment, Linux-based IP Media Server, and other components. The Pactolus SIPware Services suite adds to this a large set of carrier-read, turnkey services including prepaid and post paid calling card, conference calling, residential broadband VoIP services, voice messaging, and an operator assistance module.

PingTel (www.pingtel.com) offers the SIPxchange, a Linux-based, enterprise-class IP PBX with integrated voice mail, auto attendant, and web based configuration manager. Their Enterprise SIPxchange CallManager is a SIP communications platform with a SIP proxy and associated call routing and security modules. It supports least cost call routing and toll bypass, extends PBX functionality to mobile and remote workers when used with third party voicemail, and integrates with third party media servers (messaging, IVR, conferencing, etc.) and media gateways. Pingtel also offers a subscription-based service for its SIP Softphone which can be used SIPxchange or with third party proxies, softswitches, or carrier services.

SIP Forum (www.sipforum.org), although not a standards- setting body for SIP (that's the IETF), the SIP

Forum does have a mission to advance the adoption of SIP-based products and services. The Forum does such things as hold live interoperability test events, defines operational compliance tests, and creates white papers, implementation guides, recommendations, and other technical documents relating to SIP that fall outside the scope of the standards bodies.

SIPfoundry (www.sipfoundry.org) is a nonprofit open source community founded in February 2004 and dedicated to promoting and advancing SIP-related Open Source projects. SIPfoundry has been the center of development for sipX, the open source SIP PBX for Linux. The sipX architecture is modular and consists of a communications server, media server and configuration server. Each server can be run as a standalone.

SIPquest (www.sipquest.com) has a new deal with Nortel to enable service providers to deliver advanced SIPbased multimedia services to residential and corporate customers over wireless handheld devices.

snom (www.snom.com) recently released the snom 360 SIP telephone that supports SRTP (Secure Real Time Protocol) and SIPS (SIP-Secure) for the highest level of VoIP security. It also offers the snom 4S IP PBX, a software- based SIP server comprising the 4S Proxy for the management of user and registration data and the 4S Media Server for voicemail, auto-attendant, conferencing and other media processing applications.

U4EA Technologies (www.u4eatech.com) offers QoS and SIP software for network equipment vendors.

Ubiquity Software (www.ubiquitysoftware.com) offers SIP-based communications software for service providers, ISVs and OEMs.

Xchange Telecom (www.xchangetelecom.com) a facilities- based provider of local, long distance, calling card and prepaid services, launched a broadband telephony service called Sipmedia in August 2003 with SiPX as the foundation. The VoIP phone service is currently marketed under the myPhoneCompany brand.

Xten Networks (www.xten.com) offers SIP VoIP/Video/ IM/Presence endpoint software that supports SIMPLE, XCAP and WebDAV.

Zultys Technologies (www.zultys.com) recently debuted the new ZIP 2x2 series of phones, built entirely on open standards and running on a real-time version of Linux. The phones will interoperate with any IP telephony system using SIP. The phones support PoE (Power over Ethernet), line-rate Ethernet switching, voice encryption, and conferencing.

Zoom Technologies (www.zoom.com), which has sold modems since the 1970s, has moved on to VoIP, adding Internet phone features to its DSL ZoomTel VoIP modems and bundling them with the Global Village Internet calling service, both of which use SIP.

A VoIP Emergency Services Architecture and Prototype

Matthew Mintz-Habib, Anshuman Rawat, Henning Schulzrinne, and Xiaotao Wu
Department of Computer Science
Columbia University
{mm2571,asr,hgs,xiaotaow}@cs.columbia.edu

Abstract— **Providing emergency services in VoIP networks is vital to the success of VoIP. It not only presents design and implementation challenges, but also gives an opportunity to enhance the existing emergency call handling infrastructure. We propose an architecture to deliver emergency services in SIP-based VoIP networks, which can accommodate PSTN calls through PSTN to SIP gateways. Our architecture addresses the issues of identifying emergency calls, determining callers' locations, routing emergency calls to appropriate public safety access points (PSAP), and presenting required information to emergency call takers. We have developed a prototype implementation to prove our architecture's feasibility and scalability. We expect to undertake a pilot project at a working PSAP with our implementation once it is thoroughly tested.**

I. Introduction

VoIP telephony services are increasing in residential and enterprise communication market penetration due to their attractive service enhancements and cost savings. One feature from the traditional public switched telephone network (PSTN) that is essential for VoIP telephony is the ability to summon emergency services, such as by dialing "`911`" in the United States and "`112`" in parts of Europe. Transitioning to VoIP networks offers the opportunity to add significant enhancements to emergency call handling services, rather than simply duplicating the existing feature set. The enhancements include higher resilience, faster call setup, better information presentation, multimedia support, and lower costs. To achieve the enhancements, we designed an architecture and developed a prototype of our architecture that can provide emergency services in VoIP networks based on the Session Initiation Protocol (SIP) [1]. Our architecture can also accommodate PSTN calls bridged into VoIP networks through gateways. Even though our architecture is based on SIP, the same concepts and design principles can also be applied to other VoIP networks, such as H.323-based VoIP networks.

SIP is an application layer signaling protocol for initiating sessions between hosts to exchange media content. In SIP, sessions can be negotiated by SIP user agents (UAs) communicating directly with each other, or through a series of SIP servers, using SIP methods like INVITE, REGISTER, or BYE. Typically, SIP UAs are configured with an outbound proxy that forwards SIP messages on their behalf. SIP uses REGISTER requests to bind users' logical addresses to their physical addresses. This way, SIP can easily handle routing services, like the follow-me service, on an inbound SIP proxy server. Our architecture involves efforts on different SIP entities, including both caller and emergency call taker's user agents, and inbound and outbound SIP proxy servers.

SIP does not transport media content itself, but facilitates communicating parties to agree on what media to exchange and how to exchange it. Specifically, this is accomplished by using an offer/answer model with the Session Description Protocol (SDP) [2] as SIP message content. Note that SIP uses MIME [3] to format its content so we can put other information, such as location information, in addition to media description to form a multipart entity in SIP message content. Location information is essential for emergency call handling.

Proxy servers require emergency callers' location information to route calls to proper public safety answering points (PSAP), which are responsible for coordinating local or regional emergency services, because each PSAP is dedicated to a specific geographic area. Location information is also necessary for dispatching help to emergency callers. Since VoIP callers are nomadic, their location may not be readily apparent. Considering a user located in New York communicating through a SIP proxy in Hong Kong over a VPN tunnel. If the user were to request emergency services, the call should be routed to a call center in New York, not Hong Kong! Our architecture defines several methods to determine the location of VoIP callers.

Location information can be geographic coordinates, such as latitude, longitude, or altitude values, or civic location information, such as country, city, and street names. Civic location information needs to be general enough in an international context, since the Internet knows no national boundaries.

While there are currently no accepted standards on VoIP emergency services, related work can be found in several Internet Drafts addressing the subject [4], [5], [6]. The National Emergency Number Association's (NENA) [7], the organization promoting a universal emergency service number in the United States, recently published a list of requirements for IP enabled PSAPs [8]. Our prototype fulfills most of the requirements listed. Similarly, Arai and Kawanishi are pursuing VoIP emergency services requirements in Japan [9]. Within Columbia University, we have spent some time working on the VoIP emergency services problem [11], [12], [13], [14].

Our work brings together design features from various sources into one cohesive architecture, contributes novel design elements at the PSAP, and implements the system as a prototype. We have demonstrated our prototype implementation at working PSAPs, for local and state authorities, as well as for the members of the NENA's Next Generation E9-1-1 committee. Our demo works very well, and we expect to undertake a pilot project at a working PSAP with our implementation once it is thoroughly tested.

The remainder of our paper is organized as follows. Section II describes our emergency call handling architecture. Section III discusses challenges implementing our prototype. Section IV provides system performance and security analysis. Section V concludes the paper and discusses future work.

II. Architecture

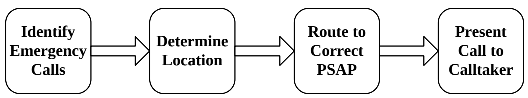


Fig. 1. Control flow for emergency call handling

Emergency call handling can be divided into four steps that are executed in sequence for each emergency call (Fig. 1). Each step involves one or more entities in the system architecture as shown in Fig. 2. The first step identifies emergency calls. For outgoing calls, the caller's user agent and outbound proxy server are responsible to check whether the call is an emergency call or not. Once an emergency call is identified, the second step determines the caller's location, and integrates the location information into call setup messages. The third step finds an appropriate PSAP based on the location information. A proxy server can then route the emergency call to the PSAP. The fourth step presents the emergency call to the emergency call taker at the PSAP. The emergency call taker utilizes the information in the call setup messages to handle the emergency call, such as pinpoint the caller on a map and bring police, fire, and medical supports into a conference call. At any point, a SIP entity may query third party services for information, such as caller location or medical records. We discuss each component in detail below.

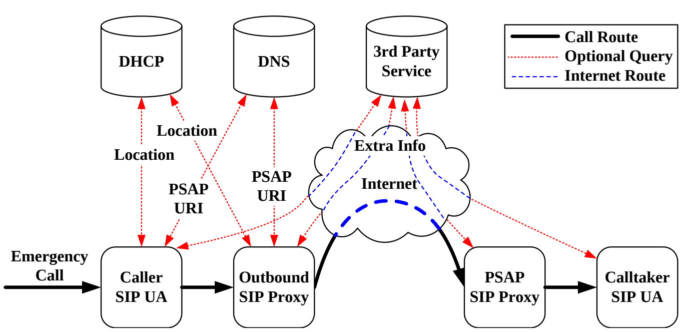


Fig. 2. Emergency services system architecture

A. Identifying emergency calls

Emergency calls are identified by their destination URIs and the location of the caller. Work is in progress to standardize the use of "`sos`" as the username part of a SIP URI [5] to represent an emergency call. Telephone URIs [15] for conventional emergency numbers, such as "`tel:911`", can be aliased to the emergency URI "`sos`", either by a SIP proxy or a SIP UA, based on the location of the caller.

B. Determining location

There are several ways to determine a calling party's location, either by the calling UA's outbound proxy or by the calling UA itself. The outbound proxy can determine the caller's location based on the calling UA's MAC address. In enterprise networks, the location of ethernet jacks and desktop machines, as well as the MAC addresses of the desktop machines are usually stored by system administrators. The outbound proxy can determine the location of the emergency caller simply by sending a `DHCPINFOM` query with the MAC address retrieved from the packets it received.

The calling UA can determine its own location directly, such as from a GPS receiver, a bluetooth beacon, DHCP options, or manually entered by a user. It can also get location information from a location server. For example, through triangulation calculation, multiple wireless access points can pinpoint a mobile station's location and store the location on a location server. The mobile station can subscribe to its own location from the location server by using SIP event notification architecture [16].

C. Routing emergency calls

Different location information require different techniques to determine appropriate PSAPs to route emergency calls to. Location information in an emergency call can be civic location, geographic coordinates, or no location. We use DNS Naming Authority Pointer Resource Records (NAPTR) [17] to find appropriate PSAPs.

To determine the correct PSAP for calls with civic location, the caller's location elements can be transformed into a period-separated form hierarchically from most granular to least granular location element. The corresponding NAPTR record has the service type "`SOS+ECC`", indicating that it represents an emergency call center's URI. DNS is first queried for the most granular location, and if no match is found, successive layers of granularity are stripped and queried until a match is found. Each location entry is suffixed with `sos-arpa.net` as the top level of the hierarchy, thus ensuring a default match if no better match is found. Upon success, a NAPTR record is returned with the emergency URI [6]. The example below shows the DNS record for the location "`Houston, TX`".

```
houston.tx.us.sos-arpa.net IN NAPTR 50 50 "u" "SOS+ECC"
"/.*/sip:houston_tx@emergency.info/i" .
```

We can also use DNS to determine the proper PSAP URI based on geographic coordinates, but with a different service type: "`SOS+POLYGON`" [6]. The result of the DNS query will be a pointer to an XML document defining a specific geo-political boundary, such as a state, county, or PSAP coverage area. These boundaries are unlikely to change often, so the

DNS record can be set with a large TTL value and the returned boundary information can be cached.

```
tx.us.sos-arpa.net IN NAPTR 50 50 "u" "SOS+POLYGON"
"/.*/http:\/\/www.emergency.info\/polygons\/texas.xml/i" .
```

The example above shows the record pointing to the XML document defining the polygon boundary for the state of Texas (special characters escaped). A proxy server can search the "SOS+POLYGON" records from least granular to most granular, linearly, to check whether a polygon contains the caller's geographic coordinates or not. Once the most granular match is found, the corresponding URI found in the "SOS+ECC" record is returned for emergency call routing.

Emergency calls that contain no location can be routed to a default PSAP URI. This URI can be determined by the outbound SIP proxy server of the calls. The proxy server queries DNS for the PSAP URI based on its own location. The default PSAP URI can also be stored as a configuration parameter in the SIP proxy.

D. Call presentation at the PSAP

A general feature list for presenting emergency calls to emergency call takers has been defined by NENA [18], which also documents features specifically for IP-enabled PSAPs [8]. PSAPs need to display caller locations on a map, automatically distribute incoming calls to available call takers, log emergency call details to database, archive call media content, view call logs and generate statistics, and monitor currently active calls. We have achieved these requirements in our prototype implementation, which we will discuss in detail below.

III. Implementation

We implemented a prototype based on the architecture defined above. To place VoIP calls, we use the Columbia SIP User Agent (SIPC) [19], as well as hardware UAs, like the Cisco 7960 [20] SIP phone. Location can be entered manually into SIPC, automatically looked up using host-specific DHCP options, received from GPS receivers, acquired from a location server through SIP event notifications, or queried through MapInfo's Envinsa [21] platform.

For call routing, we use the Columbia InterNet Extensible Multimedia Architecture (CINEMA) [10] architecture for SIP services, and use SIP-CGI [22] Perl scripts to make routing decisions. PSAP identification is accomplished by using MapInfo's Envinsa service or DNS-based lookups.

To present caller information to call takers, we use GeoComm's GeoLynx Dispatch Mapping System [23] to display caller location on a map and have SIPC interface with GeoLynx through TCP connections. Also at the PSAP level, we created a system to distribute calls among multiple call takers, enabled conferencing of multiple parties using CINEMA or the Brooktrout Technology's Snowshore Media Server [24], enhanced SIPC to log call details, and created a web-based system to manage PSAP end-systems.

A. Identifying emergency calls and determining locations

Identifying emergency calls was straightforward to implement simply by identifying calls addressed to "sos" as emergency calls. We also aliased the URIs "911" and "112" to the emergency URI for ease of use. To speed up the dialing process in an emergency, SIPC has an SOS button to quickly make emergency calls.

The more challenging aspect was determining caller location. Our prototype uses manual location entry in SIPC, though it is also capable of utilizing GPS measurement and acquiring location information from a location server. Many SIP UAs may not support manual location entry, or cannot measure or lookup their location. To accommodate such UAs, we implemented an automatic lookup feature at the outbound SIP proxy as described in Section II-B based on DHCPINFOM queries for the caller's MAC address. This ensures that every SIP UA can participate in emergency services without modification to the UA. We leave the issue of MAC address availability for calls passing through layer 3 devices as future work, though it may simply be included as a SIP header.

Complementing the DHCP lookup, the proxy can find locations for calls originating from a PSTN-to-VoIP gateway by querying the source telephone number in MapInfo's Envinsa server over an HTTP SOAP interface. This allows us to find the GPS location of cellular phones from a set of demonstration units.

Once the proxy gets the location for an incoming SIP INVITE request, it can encode the location in presence-based GEOPRIV location object format [25], and incorporate the encoded document into the message body of the INVITE request to form a multipart message body in MIME format.

These features are diagrammed in Fig. 3, which also shows the logical information flow. A user optionally enters location information into SIPC manually (1a), then makes an emergency call (1b). The call is sent to the outbound proxy (2), and may or may not include location information, depending if the location was entered manually. The outbound proxy receives the emergency call, and launches a SIP-CGI script. If necessary, the script looks up the location, either using DHCP for local callers (3a), or using MapInfo's Envinsa service for calls brought in over an IP gateway (3b). In either case, the script returns the location information (4a,4b), and further processing for routing decisions ensues.

B. Routing emergency calls

We have described the DNS-based routing strategy in Section II-C. Complimentary to the DNS lookups, we also utilize MapInfo's Envinsa platform to look up PSAP information for geographic locations. We use SIP-CGI scripts running on users' outbound proxy servers to handle emergency call routing. We allow proxy server administrators to choose Envinsa or DNS for geographic lookups by configuring the SIP-CGI scripts. If SIPC knows both civic and geographic location information, it will send both in its outgoing INVITE requests. In that case, the outbound proxy will check civic location first.

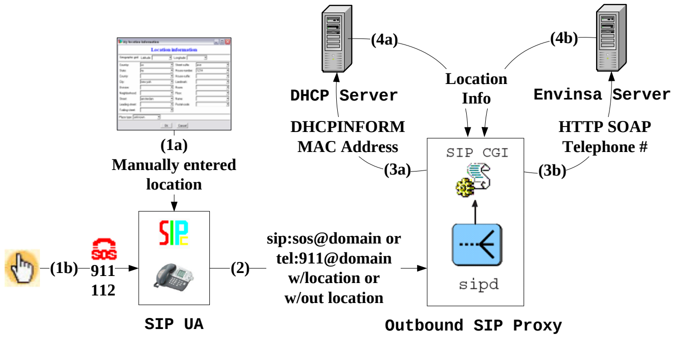


Fig. 3. Identifying emergency calls and determining location

Fig. 4 shows the overview of these features. The proxy receives an emergency call (1). If no location is available, the proxy attempts to determine the location as described in Section III-A. If location is still not available, the proxy simply routes the call to a default PSAP URI (4). If the proxy receives geographic coordinates, it will either query MapInfo's Envinsa server (2a) or DNS for PSAP boundary information (2b), depending on how the administrator of the proxy server configured the SIP-CGI script. If a civic address is received, the system queries DNS (2c) for the PSAP URI (3c). Once the SIP-CGI script get the PSAP URI (3a,3b), it will proxy the the call to the PSAP URI (4) along with the location information.

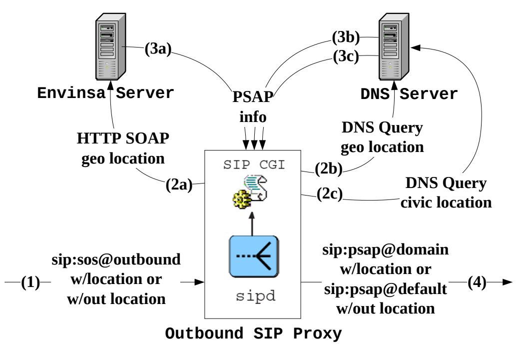


Fig. 4. Routing emergency calls

C. Call presentation at the PSAP

Once an emergency call reaches an appropriate PSAP, the PSAP UA will display the location details graphically using GeoComm's GeoLynx dispatch mapping system. When the call taker ends the session, locations are cleared from the GeoLynx display. We use SIPC as the PSAP UA, which uses TCP sockets to communicate with GeoComm's GeoLynx system. SIPC also has a button allowing emergency call takers to manually refresh location information for mobile stations using MapInfo's Envinsa platform.

SIPC has an interface to classify calls, log additional details and notes, and speed dial buttons to request police, fire, or medical support. SIPC can also transfer calls to another PSAP. For PSAPs using SIP hardphones, we implemented a mechanism for the Cisco 7960 series SIP phones to display location information, which is encoded in XML format, and retrieved via HTTP.

We use an automated controller system at the PSAP to handle all calls. The controller is responsible for distributing incoming calls to available call takers and logging the details of each call. In our prototype system, we treat every call as a conference call to allow multiple parties, including emergency call takers, police, fire, and medical support, to participate in the conference call. We have integrated two conference modules, one is CINEMA's conference server, SIPCONF, and the other is Brooktrout's Snowshore media server, both of which can be used interchangeably. The controller is responsible for managing and logging these conferences as well. Logging is performed at the earliest opportunity to provide accountability for incomplete calls.

To assist in the management of the PSAP components, we created a web interface to browse and search call logs, view call statistics, view and join active calls, update incident types, and manage associated DNS records.

Fig. 5 shows the general PSAP architecture and logical information flow. The controller receives an incoming call (1), starts logging the details (2), then creates a conference for the call (3). The controller then selects among the available call takers (4), who joins the conference in turn (5). At this point, the caller is connected to a call taker. The call taker may choose to update the caller's location information from the Envinsa Server (6a), which is then displayed in GeoLynx (7a). If necessary, the call taker may conference in additional parties such as police (6b,7b). These actions are logged (6c), and the call taker is able to log additional notes and classify the call (6c). The web management system uses the information in the datastore to generate its pages.

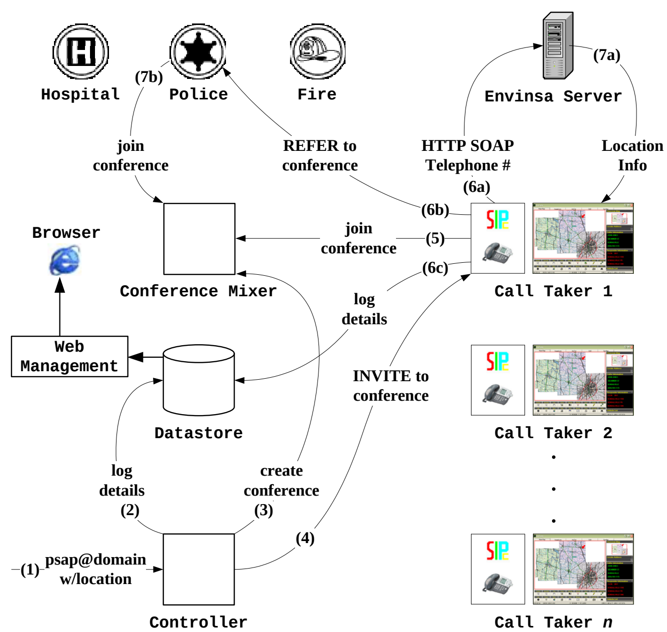


Fig. 5. PSAP architecture and logical information flow

As shown in Fig. 6, the controller uses the SIP third party call control architecture [26] to bring call takers and additional third parties in to a conference call. The process is completely transparent to participating parties. To begin, the caller initiates a SIP call, which includes location, if available. When the controller at the PSAP receives the call, it sends an INVITE to the conference server along with the caller's SDP, SDP1 (2), but without location information because the conference server is only responsible for media mixing. The conference accepts the INVITE, and sends a 200 OK with SDP1' as its message content. The controller forwards SDP1' on to the caller (4), then the controller and the caller both ACK the 200 OKs each received, respectively (5)(6). At this point, the caller is able to talk to the conference server (7).

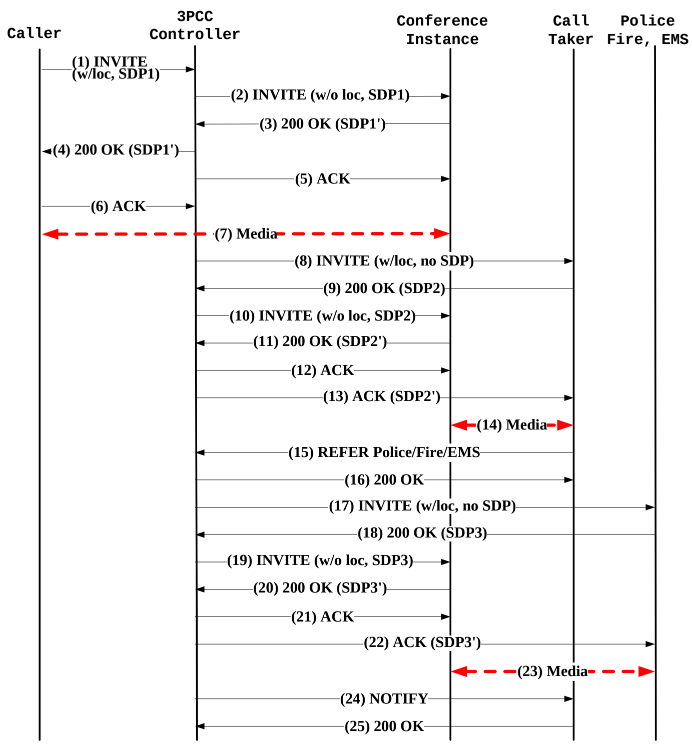


Fig. 6. Third party call control message flow

The next step is to bring a call taker into the conference to handle the emergency call. In this case, the controller sends an INVITE without an SDP body (8) so that the call taker's UA can negotiate its own media with the conference. Note that the emergency caller's location is included in the INVITE so that the call taker can immediately display the caller's location. The call taker's UA replies with a 200 OK to the INVITE and offers SDP2 (9). This offer is forwarded to the conference server in an INVITE to bring the call taker into the conference (10). The conference server accepts the INVITE and sends a 200 OK with SDP2' (11). The controller sends an ACK to the conference server (12), then puts SDP2' in its ACK to the call taker (13). Now the call taker can also talk to the conference server (14). With both the caller and call taker in the same conference, they can communicate with each other.

As is commonly the case, the call taker may want to bring in additional third parties' assistance, such as police or fire departments. We use the SIP REFER method [27] to handle this on our PSAP UA, SIPc. SIPc has speed dial buttons to bring additional parties in. Instead of sending REFER requests directly to a third party, SIPc sends the REFER requests to the controller, and has the controller to bring the third party into the conference. This way, the third party user agent does not have to support the SIP REFER method. As shown in the diagram, the call taker initiates the request to bring in a third party by sending a REFER message to the controller (15), who responds with a 200 OK (16), indicating that it is ready and able to process the request. From here, steps (17)-(23) are identical to steps (8)-(14) to bring the third party into the conference. The controller then sends a NOTIFY to the call taker (24) to update the status of the REFER request, to which the call taker responds 200 OK (25). All three parties can now communicate with each other.

IV. Performance and Security

Our prototype system has not yet undergone a comprehensive performance evaluation. The main concerns are system throughput and the latency. We define throughput as the number of emergency calls that can be handled per second, and latency as the time elapsed between emergency call initiation and the time the emergency call taker joins the call.

The throughput can be considered at the proxy level and the PSAP level. At the PSAP level, the number of simultaneous calls is most likely bounded by the number of call takers. At the proxy level, the throughput is determined by the number of requests a SIP proxy can handle. Our empirical tests have shown the CINEMA SIPd proxy running on very moderate hardware (500 MHz CPU, 128 MB RAM) capable of supporting 86 proxy requests per second [28]. More recent work shows a stateful CINEMA load-sharing architecture with failover support running on contemporary hardware (3 GHz CPU, 1 GB RAM) capable of supporting 800 calls per second. NENA estimates about 200 million 911 calls in the United States per year, or roughly 6.3 calls per second nationwide on average [7], though some emergencies may elicit a burst of calls. While the CINEMA performance evaluations did not consider SIP-CGI execution and traffic may be bursty, it is unlikely that the SIP proxy will be a bottleneck.

Latency will be a bigger concern. Many factors contribute to call setup latency, such as UA processing, network conditions, SIP proxy processing, and call distribution at the PSAP. In our prototype, much of the delay is incurred by SIP-CGI scripts waiting for queries executed on remote machines. For instance, tests on our local network show that emergency calls sent without location and routed to the default PSAP take an average of 0.57 seconds. This can be seen as a lower limit for emergency call latency in our prototype. However, calls sent with geographic location that is queried in MapInfo's Envinsa service take 1.70 seconds on average. The exact modules invoked at script run dictate the latency characteristics incurred at the SIP proxy during call setup. We will study both latency and throughput in our system as a future work.

Security considerations for our prototype implementation are less imperative than in a live, public system. Accordingly, we did not build explicit security features into our prototype. In a public system, there are some enhancements that could be added. To prevent PSAP impersonation by manipulating DNS entries, secure DNS could be used. To protect signaling integrity and media integrity and confidentiality, calls could be routed using TLS and exchange media using SRTP. Other considerations include the security involved in querying external services, such as MapInfo's Envinsa platform.

V. Conclusion

We have presented an architecture for providing emergency services in VoIP networks. Our design is based on end-to-end IP connectivity and facilitates PSTN calls bridged into the network over IP gateways. The system addresses the issues of identifying emergency calls, determining location, routing to the appropriate PSAP, and presenting the emergency call to the call taker.

The architecture was implemented into a prototype system based on Columbia University's SIP infrastructure consisting of the CINEMA platform and SIPc, as well as with components provided by MapInfo and GeoComm. We developed several software solutions to provide enhanced functionality to call takers at PSAPs, as well as provided a web-based system to manage aspects of the system.

There are many areas we are looking to explore in our prototype system. These can be grouped into the addition of new features at the PSAP, enhancements to the call delivery architecture, and performance evaluation.

At the PSAP level, we are interested in adding advanced multimedia functionality, such as playing back instructional video to callers, e.g., a CPR how-to. Another item is to implement media archiving and incorporate retrieval via the web management system. The call conference mixers we use have the ability to record audio, but no additional media types. One solution is to have an automated robot that retrieves and archives media streams join each conference call. One more useful feature we will implement is the ability to call back abandoned or disconnected emergency calls. While SIPc is capable of calling a disconnected call back directly, we currently do not support conferencing and logging of these actions.

In the call delivery architecture, we intend to add redundancy and failover features to enhance the system's robustness as described by Singh [10]. Another item is to add backup PSAP support so that if a particular PSAP's resources are occupied, incoming calls are redirected to a backup call center.

Also, we intend to conduct a comprehensive performance evaluation of the prototype system. This would empirically study both throughput and latency metrics at the system and component level.

Acknowledgments

The work described in this paper was supported in part under a grant by SIPquest and a Technology Opportunities Program (TOP) grant by the National Telecommunications and Information Administration (NTIA). The authors would also like to acknowledge Amrita Rajagopal for her contribution implementing the DNS-based XML boundary graph lookup.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session initiation protocol," RFC 3261, June 2002.

[2] J. Rosenberg and H. Schulzrinne, "An offer/answer model with the session description protocol (SDP)," RFC 3264, June 2002.

[3] N. Freed and N. Borenstein, "Multipurpose Internet mail extensions (MIME) part one: Format of Internet message bodies," RFC 2045, Nov. 1996.

[4] H. Schulzrinne and B. Rosen, "Emergency services for Internet telephony systems," draft-schulzrinne-sipping-emergency-arch-01, Internet Draft, July 2004, work in progress.

[5] H. Schulzrinne, "Emergency services URI for the session initiation protocol," draft-ietf-sipping-sos-01, Internet Draft, Feb. 2004, work in progress.

[6] B. Rosen, "Emergency call information in the domain name system," draft-rosen-dns-sos-01, Internet Draft, July 2004, work in progress.

[7] NENA, National Emergency Numbers Association. [Online]. Available: http://www.nena.org

[8] *NENA IP Capable PSAP Features And Capabilities Standard*, NENA Std. 58-001, Feb. 2005.

[9] H. Arai and M. Kawanishi, "Emergency call requirements for IP telephony services in japan," draft-arai-ecrit-japan-req-00, Internet Draft, Feb. 2005, work in progress.

[10] CINEMA, Columbia InterNet Extensible Multimedia Architecture. [Online]. Available: http://www.cs.columbia.edu/IRT/cinema/

[11] J. Lee, K. Singh, and H. Schulzrinne. SIP 911 implementation. [Online]. Available: http://www.cs.columbia.edu/~kns10/projects/spring2002/911/

[12] H. Schulzrinne and K. Arabshian, "Providing emergency services in Internet telephony," *IEEE Internet Computing*, vol. 6, no. 3, pp. 39–47, May/June 2002.

[13] X. Wu and H. Schulzrinne, "SIPc, a multi-function SIP user agent," in *IFIP/IEEE International Conference, Management of Multimedia Networks and Services (MMNS'04)*, San Diego, CA, Oct. 2004, pp. 269–281.

[14] ——, "Location-based services in Internet telephony," in *IEEE Consumer Communications & Networking Conference (CCNC'05)*, Las Vegas, NV, Jan. 2005.

[15] H. Schulzrinne, "The tel URI for telephone numbers," RFC 3966, Dec. 2004.

[16] A. B. Roach, "Session initiation protocol (SIP)-specific event notification," RFC 3265, June 2002.

[17] M. Mealling and R. W. Daniel, "The naming authority pointer (NAPTR) DNS resource record," RFC 2915, Sept. 2000.

[18] *NENA Generic E9-1-1 Requirements Technical Information Document*, NENA TID 08-502, July 2004.

[19] sipc, Columbia SIP User Agent. [Online]. Available: http://www1.cs.columbia.edu/~xiaotaow/sipc/

[20] Cisco Systems. VoIP phones. [Online]. Available: http://www.cisco.com

[21] MapInfo Corporation. Envinsa Location Platform. [Online]. Available: http://www.mapinfo.com

[22] J. Lennox, H. Schulzrinne, and J. Rosenberg, "Common gateway interface for SIP," RFC 3050, Jan. 2001.

[23] GeoComm Corporation. GeoLynx Dispatch Mapping System. [Online]. Available: http://www.geo-comm.com

[24] Brooktrout Technology. Snowshore Media Server. [Online]. Available: http://www.brooktrout.com

[25] J. Peterson, "A presence-based GEOPRIV location object format," draft-ietf-geopriv-pidf-lo-03, Internet Draft, Sept. 2004, work in progress.

[26] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, "Best current practices for third party call control (3pcc) in the session initiation protocol (SIP)," RFC 3725, Apr. 2004.

[27] R. Sparks, "The session initiation protocol (SIP) refer method," RFC 3515, Apr. 2003.

[28] J. Lennox, "Services for Internet telephony," Ph.D. dissertation, Department of Computer Science, Columbia University, New York, New York, 2004, pp.113-117.

The emerging H.264/AVC standard

Ralf Schäfer, Thomas Wiegand and Heiko Schwarz
Heinrich Hertz Institute, Berlin, Germany

H.264/AVC is the current video standardization project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The main goals of this standardization effort are to develop a simple and straightforward video coding design, with enhanced compression performance, and to provide a "network-friendly" video representation which addresses "conversational" (video telephony) and "non-conversational" (storage, broadcast or streaming) applications.

H.264/AVC has achieved a significant improvement in the rate-distortion efficiency – providing, typically, a factor of two in bit-rate savings when compared with existing standards such as MPEG-2 Video.

The MPEG-2 video coding standard [1], which was developed about 10 years ago, was the enabling technology for all digital television systems worldwide. It allows an efficient transmission of TV signals over satellite (DVB-S), cable (DVB-C) and terrestrial (DVB-T) platforms. However, other transmission media such as xDSL or UMTS offer much smaller data rates. Even for DVB-T, there is insufficient spectrum available – hence the number of programmes is quite limited, indicating a need for further improved video compression.

In 1998, the *Video Coding Experts Group* (VCEG – ITU-T SG16 Q.6) started a project called H.26L with the target to double the coding efficiency when compared with any other existing video coding standard. In December 2001, VCEG and the *Moving Pictures Expert Group* (MPEG – ISO/IEC JTC 1/SC 29/WG 11) formed the *Joint Video Team* (JVT) with the charter to finalize the new video coding standard H.264/AVC [2].

The H.264/AVC design covers a **Video Coding Layer** (VCL), which efficiently represents the video content, and a **Network Abstraction Layer** (NAL), which formats the VCL representation of the video and provides header information in a manner appropriate for conveyance by particular transport layers or storage media.

The VCL design – as in any prior ITU-T and ISO/IEC JTC1 standard since H.261 [2] – follows the so-called *block-based hybrid* video-coding approach. The basic source-coding algorithm is a hybrid of *inter-picture prediction*, to exploit the temporal statistical dependencies, and *transform coding of the prediction residual* to exploit the spatial statistical dependencies. There is no single coding element in the VCL that provides the majority of the dramatic improvement in compression efficiency, in relation to prior video coding standards. Rather, it is the plurality of smaller improvements that add up to the significant gain.

The next section provides an overview of the H.264/AVC design. The Profiles and Levels specified in the current version of H.264/AVC [2] are then briefly described, followed by a comparison of H.264/AVC Main profile with the profiles of prior coding standards, in terms of rate-distortion efficiency. Based on the study of rate-distortion performance, various new business opportunities are delineated, followed by a report on existing implementations.

Technical overview of H.264/AVC

The H.264/AVC design [2] supports the coding of video (in 4:2:0 chroma format) that contains either progressive or interlaced frames, which may be mixed together in the same sequence. Generally, a frame of video contains two interleaved fields, the top and the bottom field. The two fields of an interlaced frame, which are separated in time by a field period (half the time of a frame period), may be coded separately as two field pictures or together as a frame picture. A progressive frame should always be coded as a single frame picture; however, it is still considered to consist of two fields at the same instant in time.

Network abstraction layer

The VCL, which is described in the following section, is specified to represent, efficiently, the content of the video data. The NAL is specified to format that data and provide header information in a manner appropriate for conveyance by the transport layers or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and bitstream delivery is identical – except that each NAL unit can be preceded by a start code prefix in a bitstream-oriented transport layer.

Video coding layer

The video coding layer of H.264/AVC is similar in spirit to other standards such as MPEG-2 Video. It consists of a hybrid of temporal and spatial prediction, in conjunction with transform coding. *Fig. 1* shows a block diagram of the video coding layer for a macroblock.

In summary, the picture is split into blocks. The first picture of a sequence or a random access point is typically “Intra” coded, i.e., without using information other than that contained in the picture itself. Each sample

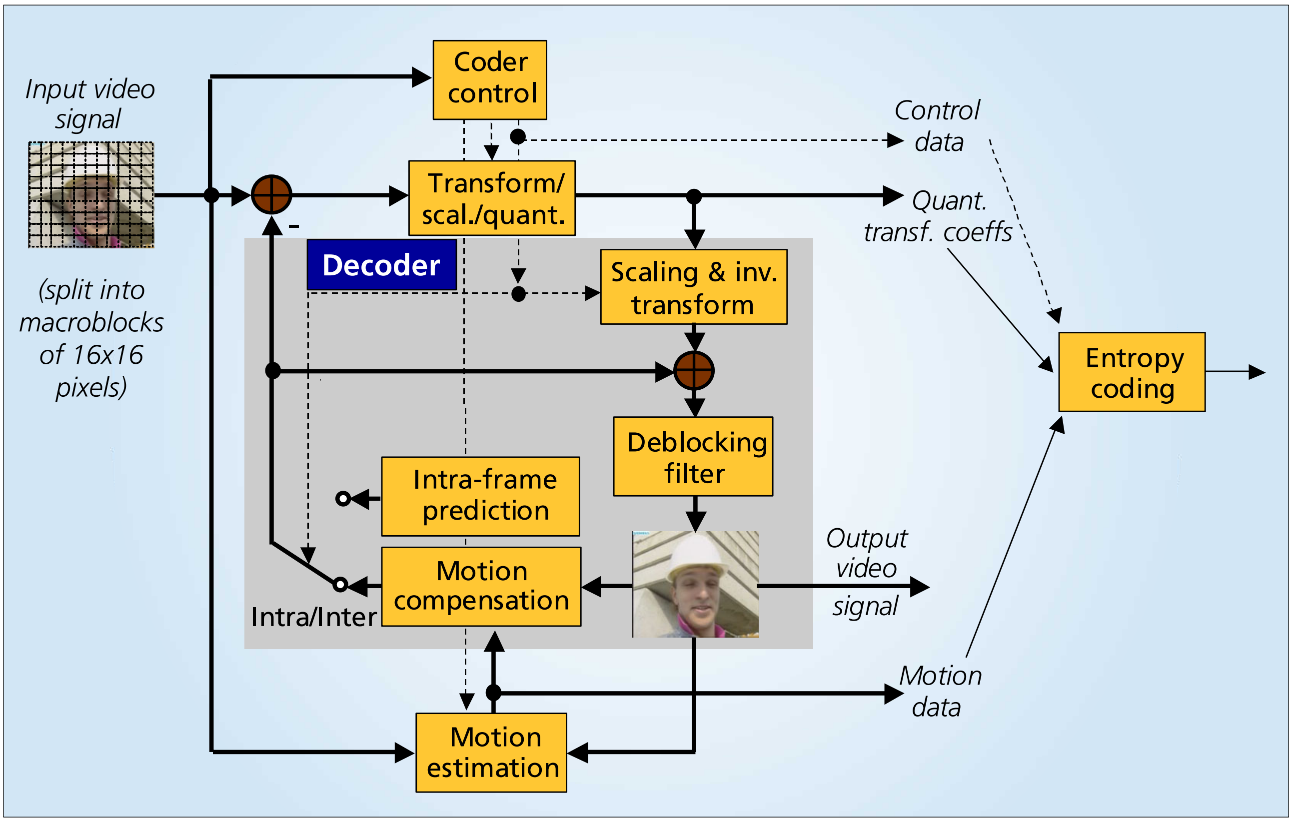


Figure 1
Basic coding structure of H.264/AVC for a macroblock

of a block in an Intra frame is predicted using spatially neighbouring samples of previously coded blocks. The encoding process chooses which and how neighbouring samples are used for Intra prediction, which is simultaneously conducted at the encoder and decoder using the transmitted Intra prediction side information.

For all remaining pictures of a sequence or between random access points, typically "Inter" coding is used. Inter coding employs prediction (motion compensation) from other previously decoded pictures. The encoding process for Inter prediction (motion estimation) consists of choosing motion data, comprising the reference picture, and a spatial displacement that is applied to all samples of the block. The motion data which are transmitted as side information are used by the encoder and decoder to simultaneously provide the Inter prediction signal.

The residual of the prediction (either Intra or Inter) – which is the difference between the original and the predicted block – is transformed. The transform coefficients are scaled and quantized. The quantized transform coefficients are entropy coded and transmitted together with the side information for either Intra-frame or Inter-frame prediction.

The encoder contains the decoder to conduct prediction for the next blocks or the next picture. Therefore, the quantized transform coefficients are inverse scaled and inverse transformed in the same way as at the decoder side, resulting in the decoded prediction residual. The decoded prediction residual is added to the prediction. The result of that addition is fed into a deblocking filter which provides the decoded video as its output.

A more detailed description of the technical contents of H.264 is given below. Readers less interested in technical details may want to skip these sections and continue by reading the section on "Profiles and levels" *(see page 8)*.

Subdivision of a picture into macroblocks

Each picture of a video, which can either be a frame or a field, is partitioned into fixed-size macroblocks that cover a rectangular picture area of 16×16 samples of the luma component and 8×8 samples of each of the two

Abbreviations

3G 3rd Generation mobile communications
3GPP 3rd Generation Partnership Project
16-QAM 16-state Quadrature Amplitude Modulation
ASP (MPEG-4) Advanced Simple Profile
CABAC Context-Adaptive Binary Arithmetic Coding
CAVLC Context-Adaptive Variable Length Coding
CIF Common Intermediate Format
DCT Discrete Cosine Transform
DVB Digital Video Broadcasting
DVB-C DVB - Cable
DVB-S DVB - Satellite
DVB-T DVB - Terrestrial
FIR Finite Impulse Response
FMO Flexible Macroblock Ordering
FPGA Field-Programmable Gate Array
IBC International Broadcasting Convention
IEC International Electrotechnical Commission
ISO International Organization for Standardization
ITU International Telecommunication Union
ITU-T ITU - Telecommunication Standardization Sector
JTC (ISO/IEC) Joint Technical Committee
JVT (MPEG/VCEG) Joint Video Team
HLP (H.263++) High Latency Profile
MP@ML (MPEG-2) Main Profile at Main Level
MPEG (ISO/IEC) Moving Picture Experts Group
NAL Network Abstraction Layer
PAL Phase Alternation Line
PSNR Peak Signal-to-Noise Ratio
QAM Quadrature Amplitude Modulation
QCIF Quarter Common Intermediate Format
QP Quantization Parameter
QPSK Quadrature (Quaternary) Phase-Shift Keying
SRAM Static Random Access Memory
UMTS Universal Mobile Telecommunication System
VCEG (ITU-T) Video Coding Experts Group
VCL Video Coding Layer
xDSL *(Different variants of)* Digital Subscriber Line

chroma components. All luma and chroma samples of a macroblock are either spatially or temporally predicted, and the resulting prediction residual is transmitted using transform coding. Therefore, each colour component of the prediction residual is subdivided into blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and transmitted using entropy-coding methods.

The macroblocks are organized in slices, which generally represent subsets of a given picture that can be decoded independently. The transmission order of macroblocks in the bitstream depends on the so-called *Macroblock Allocation Map* and is not necessarily in raster-scan order. H.264/AVC supports five different slice-coding types. The simplest one is the **I** slice (where "I" stands for intra). In I slices, all macroblocks are coded without referring to other pictures within the video sequence. On the other hand, prior-coded images can be used to form a prediction signal for macroblocks of the predictive-coded **P** and **B** slices (where "P" stands for predictive and "B" stands for bi-predictive).

The remaining two slice types are **SP** (switching P) and **SI** (switching I), which are specified for efficient switching between bitstreams coded at various bit-rates. The Inter prediction signals of the bitstreams for one selected SP frame are quantized in the transform domain, forcing them into a coarser range of amplitudes. This coarser range of amplitudes permits a low bit-rate coding of the difference signal between the bitstreams. SI frames are specified to achieve a perfect match for SP frames in cases where Inter prediction cannot be used because of transmission errors.

In order to provide efficient methods for concealment in error-prone channels with low delay applications, a feature called *Flexible Macroblock Ordering* (FMO) is supported by H.264/AVC. FMO specifies a pattern that assigns the macroblocks in a picture to one or several slice groups. Each slice group is transmitted separately. If a slice group is lost, the samples in spatially neighbouring macroblocks that belong to other correctly-received slice groups can be used for efficient error concealment. The allowed patterns range from rectangular patterns to regular scattered patterns, such as chess boards, or to completely random scatter patterns.

Intra-frame prediction

Each macroblock can be transmitted in one of several coding types depending on the slice-coding type. In all slice-coding types, two classes of intra coding types are supported, which are denoted as INTRA-4×4 and INTRA-16×16 in the following. In contrast to previous video coding standards where prediction is conducted in the transform domain, prediction in H.264/AVC is always conducted in the spatial domain by referring to neighbouring samples of already coded blocks.

When using the INTRA-4×4 mode, each 4×4 block of the luma component utilizes one of nine prediction modes. Beside DC prediction, eight directional prediction modes are specified. When utilizing the INTRA-16×16 mode, which is well suited for smooth image areas, a uniform prediction is performed for the whole luma component of a macroblock. Four prediction modes are supported. The chroma samples of a macroblock are always predicted using a similar prediction technique as for the luma component in Intra-16x16 macroblocks. Intra prediction across slice boundaries is not allowed in order to keep all slices independent of each other.

Motion compensation in P slices

In addition to the Intra macroblock coding types, various predictive or motion-compensated coding types are specified for P-slice macroblocks. Each P-type macroblock corresponds to a specific partitioning of the macroblock into fixed-size blocks used for motion description. Partitions with luma block sizes of 16×16, 16×8, 8×16 and 8×8 samples are supported by the syntax corresponding to the Inter-16×16, Inter-16×8, Inter-8×16 and Inter-8×8 P macroblock types, respectively. In cases where the Inter-8×8 macroblock mode is chosen, one additional syntax element for each 8×8 sub-macroblock is transmitted. This syntax element specifies if the corresponding sub-macroblock is coded using motion-compensated prediction with luma block sizes of 8×8, 8×4, 4×8 or 4×4 samples. *Fig. 2* illustrates the partitioning.

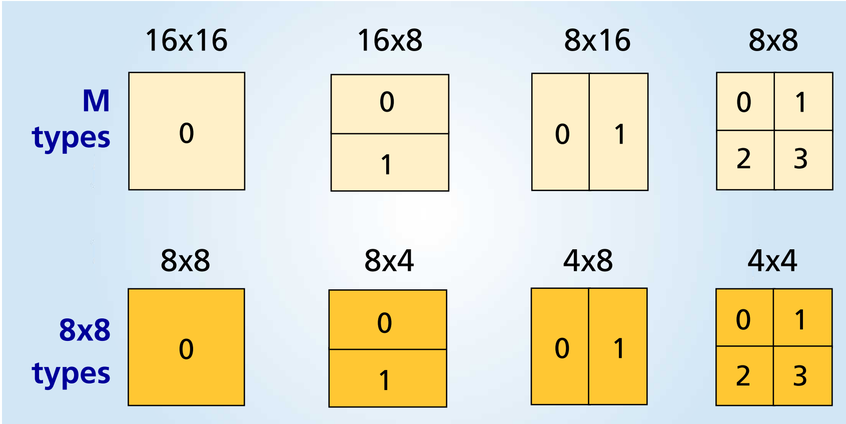


Figure 2
Segmentations of the macroblock for motion compensation.
***Top:* segmentation of macroblocks.**
***Bottom:* segmentation of 8x8 partitions.**

The prediction signal for each predictive-coded m×n luma block is obtained by displacing an area of the corresponding reference picture, which is specified by a translational motion vector and a picture reference index. Thus, if the macroblock is coded using the Inter-8x8 macroblock type, and each sub-macroblock is coded using the Inter-4x4 sub-macroblock type, a maximum of sixteen motion vectors may be transmitted for a single P-slice macroblock.

The accuracy of motion compensation is a quarter of a sample distance. In cases where the motion vector points to an integer-sample position, the prediction signals are the corresponding samples of the reference picture; otherwise, they are obtained by using interpolation at the sub-sample positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter. Prediction values at quarter-sample positions are generated by averaging samples at the integer- and half-sample positions.

The prediction values for the chroma components are always obtained by bi-linear interpolation.

The H.264/AVC syntax generally allows unrestricted motion vectors, i.e. motion vectors can point outside the image area. In this case, the reference frame is extended beyond the image boundaries by repeating the edge pixels before interpolation. The motion vector components are differentially coded using either median or directional prediction from neighbouring blocks. No motion vector component prediction takes place across slice boundaries.

H.264/AVC supports multi-picture motion-compensated prediction. That is, more than one prior-coded picture can be used as a reference for motion-compensated prediction. *Fig. 3* illustrates the concept.

Both the encoder and decoder have to store the reference pictures used for Inter-picture prediction in a multi-picture buffer. The decoder replicates the multi-picture buffer of the encoder, according to the reference picture buffering type and any memory management control operations that are specified in the bitstream. Unless the size of the multi-picture buffer is set to one picture, the index at which the reference picture is located inside the multi-picture buffer has to be signalled. The reference index parameter is transmitted for each motion-compensated 16×16, 16×8, 8×16 or 8x8 luma block.

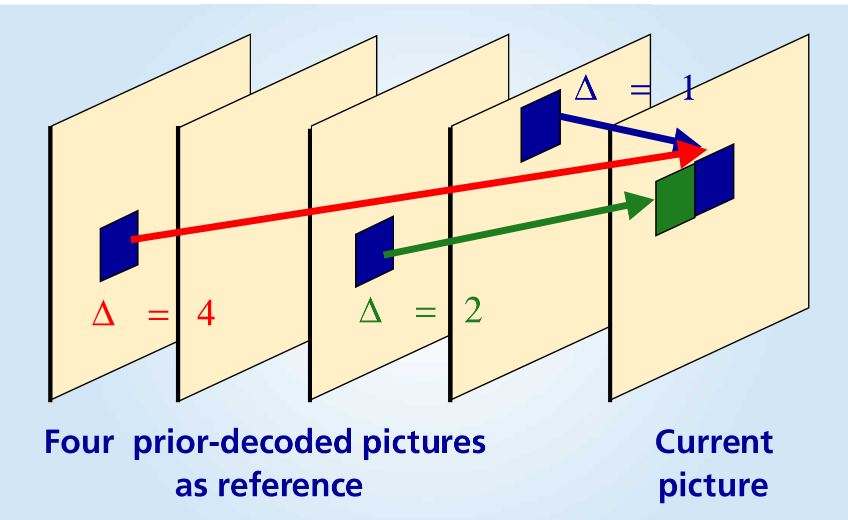


Figure 3
Multi-frame motion compensation. In addition to the motion vector, also picture reference parameters (Δ) are transmitted. The concept is also extended to B pictures as described below.

In addition to the motion-compensated macroblock modes described above, a P-slice macroblock can also be coded in the so-called SKIP mode. For this mode, neither a quantized prediction error signal, nor a motion vector or reference index parameter, has to be transmitted. The reconstructed signal is obtained in a similar way to the prediction signal of an Inter-16×16 macroblock that references the picture, which is located at index 0 in the multi-picture buffer. In general, the motion vector used for reconstructing the SKIP macroblock is identical to the motion vector predictor for the 16×16 block. However, if special conditions hold, a zero motion vector is used instead.

Motion compensation in B slices

In comparison to prior video-coding standards, the concept of B slices is generalized in H.264/AVC. For example, other pictures can reference B pictures for motion-compensated prediction, depending on the memory management control operation of the multi-picture buffering. Thus, the substantial difference between B and P slices is that B slices are coded in a manner in which some macroblocks or blocks may use a weighted average of two distinct motion-compensated prediction values, for building the prediction signal. Generally, B slices utilize two distinct reference picture buffers, which are referred to as the *first* and *second* reference picture buffer, respectively. Which pictures are actually located in each reference picture buffer is an issue for the multi-picture buffer control, and an operation very similar to the well-known MPEG-2 B pictures can be enabled.

In B slices, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive, and direct prediction. While list 0 prediction indicates that the prediction signal is formed by utilizing motion compensation from a picture of the first reference picture buffer, a picture of the second reference picture buffer is used for building the prediction signal if list 1 prediction is used. In the bi-predictive mode, the prediction signal is formed by a weighted average of a motion-compensated list 0 and list 1 prediction signal. The direct prediction mode is inferred from previously transmitted syntax elements and can be either list 0 or list 1 prediction or bi-predictive.

B slices utilize a similar macroblock partitioning to P slices. Besides the Inter-16×16, Inter-16×8, Inter-8×16, Inter-8×8 and the Intra modes, a macroblock type that utilizes direct prediction, i.e. the direct mode, is provided. Additionally, for each 16×16, 16×8, 8×16, and 8×8 partition, the prediction method (list 0, list 1, bi-predictive) can be chosen separately. An 8×8 partition of a B-slice macroblock can also be coded in direct mode. If no prediction error signal is transmitted for a direct macroblock mode, it is also referred to as *B slice SKIP mode* and can be coded very efficiently, similar to the SKIP mode in P slices. The motion vector coding is similar to that of P slices with the appropriate modifications because neighbouring blocks may be coded using different prediction modes.

Transform, scaling and quantization

Similar to previous video coding standards, H.264/AVC also utilizes transform coding of the prediction residual. However, in H.264/AVC, the transformation is applied to 4×4 blocks, and instead of a 4×4 discrete cosine transform (DCT), a separable integer transform – with basically the same properties as a 4×4 DCT – is used. Since the inverse transform is defined by exact integer operations, inverse-transform mismatches are avoided. An additional 2×2 transform is applied to the four DC coefficients of each chroma component. If a macroblock is coded in Intra-16x16 mode, a similar 4x4 transform is performed for the 4x4 DC coefficients of the luma signal. The cascading of block transforms is equivalent to an extension of the length of the transform functions.

For the quantization of transform coefficients, H.264/AVC uses scalar quantization. One of 52 quantizers is selected for each macroblock by the Quantization Parameter (QP). The quantizers are arranged so that there is an increase of approximately 12.5% in the quantization step size when incrementing the QP by one. The quantized transform coefficients of a block are generally scanned in a zigzag fashion and transmitted using entropy coding methods. For blocks that are part of a macroblock coded in field mode, an alternative scanning pattern is used. The 2×2 DC coefficients of the chroma component are scanned in raster-scan order. All transforms in H.264/AVC can be implemented using only additions to, and bit-shifting operations on, the 16-bit integer values.

Entropy coding

In H.264/AVC, two methods of entropy coding are supported. The default entropy coding method uses a single infinite-extend codeword set for all syntax elements, except the quantized transform coefficients. Thus, instead of designing a different VLC table for each syntax element, only the mapping to the single codeword table is customized according to the data statistics. The single codeword table chosen is an exp-Golomb code with very simple and regular decoding properties.

For transmitting the quantized transform coefficients, a more sophisticated method called Context-Adaptive Variable Length Coding (CAVLC) is employed. In this scheme, VLC tables for various syntax elements are switched, depending on already-transmitted syntax elements. Since the VLC tables are well designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using just a single VLC table.

The efficiency of entropy coding can be improved further if Context-Adaptive Binary Arithmetic Coding (CABAC) is used [3]. On the one hand, the use of arithmetic coding allows the assignment of a non-integer number of bits to each symbol of an alphabet, which is extremely beneficial for symbol probabilities much greater than 0.5. On the other hand, the use of adaptive codes permits adaptation to non-stationary symbol statistics. Another important property of CABAC is its context modelling. The statistics of already-coded syntax elements are used to estimate the conditional probabilities. These conditional probabilities are used for switching several estimated probability models. In H.264/AVC, the arithmetic coding core engine and its associated probability estimation are specified as multiplication-free low-complexity methods, using only shifts and table look-ups. Compared to CAVLC, CABAC typically provides a reduction in bit-rate of between 10 - 15% when coding TV signals at the same quality.

In-loop deblocking filter

One particular characteristic of block-based coding is visible block structures. Block edges are typically reconstructed with less accuracy than interior pixels and “blocking” is generally considered to be one of the most visible artefacts with the present compression methods. For this reason H.264/AVC defines an adaptive in-loop deblocking filter, where the strength of filtering is controlled by the values of several syntax elements. The blockiness is reduced without much affecting the sharpness of the content. Consequently, the subjective quality is significantly improved. At the same time the filter reduces bit-rate with typically 5-10% while producing the same objective quality as the non-filtered video.

Fig. 4 illustrates the performance of the deblocking filter.

Figure 4
Performance of the deblocking filter for highly compressed pictures.
***Left:* without the deblocking filter. *Right:* with the deblocking filter.**

Interlace coding tools

Frames can be coded as one unit or can be split into two fields which can be coded as separate units again. This field coding is especially efficient if the first field is coded using I slices and the second field makes a prediction from it using motion compensation. Furthermore, field coding is often utilized when the scene shows strong horizontal motion.

In some scenarios, parts of the frame are more efficiently coded in field mode while other parts are more efficiently coded in frame mode. Hence, H.264/AVC supports macroblock-adaptive switching between frame and field coding. For that, a pair of vertically connected macroblocks is coded as two frame or field macroblocks. The prediction processes and prediction residual coding is then either conducted assuming a frame, or field to be coded. The deblocking filtering takes place for all macroblock pairs when they are put into the frame in frame mode, regardless of whether they have been coded in frame or field mode.

Profiles and levels

Profiles and levels specify the conformance points. These conformance points are designed to facilitate interoperability between various applications of the H.262/AVC standard that have similar functional requirements. A *profile* defines a set of coding tools or algorithms that can be used in generating a compliant bitstream, whereas a *level* places constraints on certain key parameters of the bitstream.

All decoders conforming to a specific profile have to support all features in that profile. Encoders are not required to make use of any particular set of features supported in a profile but have to provide conforming bitstreams. In H.264/AVC, three profiles are defined – Baseline, Main and X:

- The **Baseline** profile supports all features in H.264/AVC except the following two feature sets:
  - **Set 1**: B slices, weighted prediction, CABAC, field coding and macroblock adaptive switching between frame and field coding.
  - **Set 2**: SP and SI slices.
- The first set of features is supported by **Main** profile. However, Main profile does not support the FMO feature which is supported by the Baseline profile.
- **Profile X** supports both sets of features on top of the Baseline profile, except for CABAC and macroblock adaptive switching between frame and field coding.

In H.264/AVC, the same set of level definitions is used with all profiles, but individual implementations may support a different level for each supported profile. Eleven levels are defined, specifying upper limits for the picture size (in macroblocks), the decoder-processing rate (in macroblocks per second), the size of the multi-picture buffers, the video bit-rate and the video buffer size.

Comparison of H.264/AVC coding efficiency with that of prior coding standards

For demonstrating the coding performance of H.264/AVC [2], we compared it to the successful prior coding standards MPEG-2 Visual [1], H.263++ [3], and MPEG-4 Visual [4] for a set of popular QCIF (10 Hz and 15 Hz) and CIF (15 Hz and 30 Hz) sequences with different motion and spatial detail information. The QCIF sequences were: *Foreman*, *News*, *Container Ship* and *Tempete*. The CIF sequences were: *Bus*, *Flower Garden*, *Mobile and Calendar* and *Tempete*. Based on [5][6], all video encoders were optimized with regards to their rate-distortion efficiency using Lagrangian techniques. In addition to the performance gains, the use of a unique and efficient coder control for all video encoders allowed a fair comparison between them in terms of coding efficiency.

During these tests, the MPEG-2 Visual encoder generated bitstreams at the well-known MP@ML conformance point, and the H.263++ encoder used the features of the High Latency Profile (HLP). In the case of

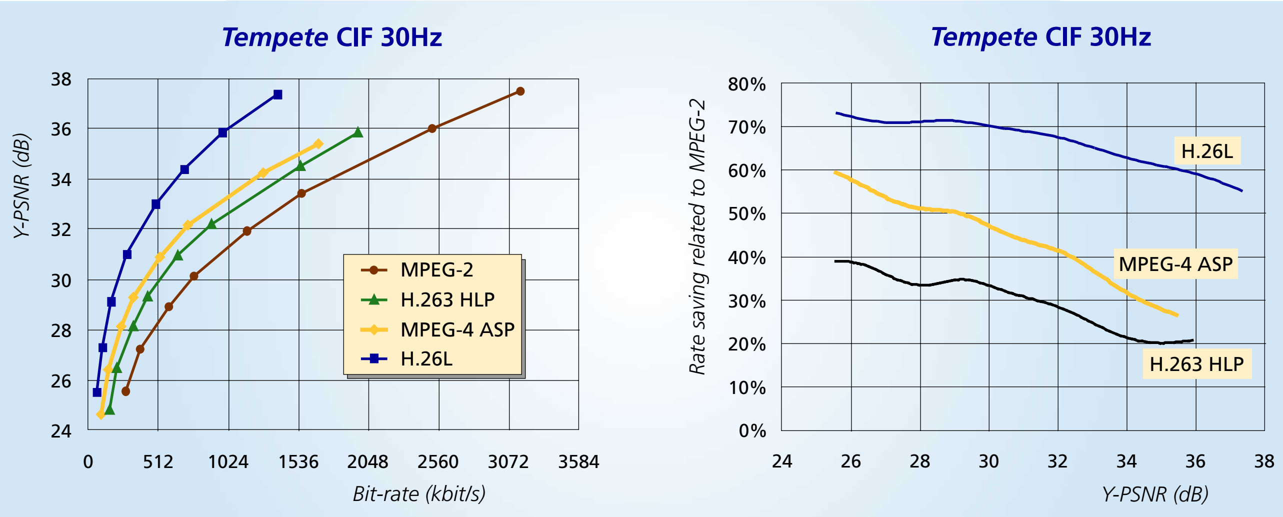


Figure 5
Selected rate-distortion curves and bit-rate saving plots

MPEG-4 Visual, the Advanced Simple Profile (ASP) was used with quarter-sample-accurate motion compensation and global motion compensation enabled. Additionally, the recommended deblocking/deringing filter was applied as a post-processing operation.

For the H.264/AVC JM-2.0 coder, the features enabled in the Main profile were used. We generally used five reference frames for both H.263 and H.264/AVC, with the exception of the News sequences where we used more reference frames for exploiting the known redundancies within this special sequence. With all the coders under test, only the first picture of each sequence was coded as an I-picture, and two B-pictures were inserted between two successive P-pictures. For H.264/AVC, the B-pictures were not stored in the multi-picture buffer, and thus the following pictures did not reference them. Full search motion estimation, with a range of 32 integer pixels, was used by all the encoders along with the Lagrangian coder control from [5][6]. The bit-rates were adjusted by using a fixed quantization parameter.

Fig. 5 shows the rate-distortion curves of all four codecs, for the sequence *Tempete* in CIF resolution.

On the right-hand chart in *Fig. 5*, the bit-rate saving relative to the worst tested video coding standard, MPEG-2, is plotted against the PSNR of the luma component for H.263 HLP, MPEG-2 ASP and H.264/AVC (marked as H.26L). The average bit-rate savings provided by each encoder, relative to all other tested encoders over the entire set of sequences and bit-rates, are depicted in *Table 1*. It can be seen that H.264/AVC significantly outperforms all other standards. The highly flexible motion model and the very efficient context-based arithmetic-coding scheme are the two primary factors that enable the superior rate-distortion performance of H.264/AVC.

Table 1
Average bit-rate savings compared with various prior decoding schemes

Coder	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC	38.62%	48.80%	64.46%
MPEG-4 ASP	-	16.65%	42.95%
H.263 HLP	-	-	30.61%

Although not discussed in this article, the bit-rates for TV or HD video (at broadcast and DVD quality) are reduced by a factor of between 2.25 and 2.5 – when using H.264/AVC coding.

New application areas and business models

The increased compression efficiency of H.264/AVC offers new application areas and business opportunities. It is now possible, to transmit video signals at about 1 Mbit/s with TV (PAL) quality, which enables streaming over xDSL connections. Another interesting business area is TV transmission over satellite. By choosing 8-PSK and turbo coding (as currently under discussion for DVB-S2) and the usage of H.264/AVC, the number of programmes per satellite can be tripled in comparison to the current DVB-S systems using MPEG-2. Given this huge amount of additional transmission capacity, even the exchange of existing set-top boxes might become an interesting option.

Also for DVB-T, H.264/AVC is an interesting option. Assuming the transmission parameters which have been selected for Germany (8k mode, 16-QAM, code rate 2/3, and ¼ Guard Interval), a bitrate of 13.27 Mbit/s is available in each 8 MHz channel. Using MPEG-2 coding, the number of TV programmes per channel is restricted to four whereas, by using H.264/AVC, the number of programmes could be raised to ten or even more, because not only the coding efficiency but also the statistical multiplex gain for variable bit-rates is higher due to the higher number of different programmes. Another interesting option, relating to the discussions on “electro-smog”, is to use QPSK, code rate ½ in conjunction with H.264/AVC. This combination would allow us to retain four programmes per channel, but to decrease the transmitted power by 15% in comparison to the transmission mode mentioned above (16 QAM, 2/3).

A further interesting business area is HD transmission and storage. It now becomes possible to encode HD signals at about 8 Mbit/s which fit onto a conventional DVD. This will surely stimulate and accelerate the home cinema market, because it is no longer necessary to wait for the more expensive and unreliable blue DVD laser. It is also possible to transmit 4 HD programs per satellite or cable channel, which makes this service much more attractive to broadcasters, as the transmission costs are much lower than with MPEG-2.

Also in the field of mobile communication, H.264/AVC will play an important role because the compression efficiency will be doubled in comparison to the coding schemes currently specified by 3GPP for streaming [7], i.e. H.263 Baseline, H.263+ and MPEG-4 Simple Profile. This is extremely important because the data rate available in 3G systems works out to be very expensive.

Implementation reports

The H.264/AVC standard only specifies the decoder, as this has been the usual procedure for all other international video coding standards before. Therefore, the rate-distortion performance and complexity of the encoder is up to the manufacturers. Nevertheless, the JVT always requests – for every decoder feature that is proposed – an example encoding method that demonstrates the feasibility of usage of that feature, together with the associated benefits. If the feature is adopted, the proponent is requested to integrate it into the reference software. During the development of H.264/AVC, about 100 proposals from 20 different companies have been integrated into the reference software, making this piece of software very slow and not usable for practical implementation. Therefore, complexity analysis – based on the reference software, e.g., as reported in [8] – typically overstates the actual complexity of the H.264/AVC encoder (by an order of magnitude) and that of the decoder (by a factor of 2 - 3).

In September 2002, at IBC in Amsterdam, VideoLocus showed a demo consisting of its own highly-optimized H.264/AVC codec, running a DVD-quality video stream at 1 Mbits/s in a side-by-side comparison with an MPEG-2 video stream at 5 Mbits/s. VideoLocus’ encoder algorithms run on a Pentium 4 platform with hardware acceleration coming from an add-in FPGA card which performs motion estimation, estimation of Intra-prediction, mode decision statistics and video-preprocessing support [9].

In October 2002, UBVideo [10] showed (for the H.264/AVC Baseline profile) CIF-resolution video running on a 800 MHz Pentium 3 laptop computer. The encoding was at 49 frames per second (fps), decoding at 105 fps, and encoding and decoding together at 33 fps. Their low-complexity encoding solution – which is designed/optimized for real-time conversational video applications – incurred an increase in bit-rate of approximately 10% against the rate-distortion performance of the very slow reference software, when encoding typical video content used in such applications.

Like many other companies including Deutsche Telekom, Broadcom, Nokia or Motorola, the Heinrich Hertz Institute (in Berlin, Germany) is developing H.264/AVC real-time solutions. A software implementation, running on a Pentium 4 platform, achieves real-time TV-resolution decoding and 20 Hz CIF encoding with less than 10 - 15 % bit-rate increase over the rate-distortion performance of the very slow reference software. HHI's decoder implementation has been ported on an ARM922 processor, running at 200 MHz, SRAM, showing 6 fps video at CIF resolution and 25 fps video at QCIF resolution.

Conclusions

H.264/AVC represents a major step forward in the development of video coding standards. It typically outperforms all existing standards by a factor of two and especially in comparison to MPEG-2, which is the basis for digital TV systems worldwide; an improvement factor of 2.25 - 2.5 has been reached. This improvement enables new applications and business opportunities to be developed. Example uses for DVB-T, DVB-S2, DVD, xDSL and 3G have been presented. Although H.264/AVC is 2 -3 times more complex than MPEG-2 at the decoder and 4 - 5 times more complex at the encoder, it is relatively less complex than MPEG-2 was at its outset, due to the huge progress in technology which has been made since then.

Another important fact is that H.264/AVC is a public and open standard. Every manufacturer can build encoders and decoders in a competitive market. This will bring prices down quickly, making this technology affordable to everybody. There is no dependency on proprietary formats, as on the Internet today, which is of utmost importance for the broadcast community.

Bibliography

[1] ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2): **Generic coding of moving pictures and associated audio information – Part 2: Video**
ITU-T and ISO/IEC JTC1, November 1994.

[2] T. Wiegand: **Joint Final Committee Draft**
Doc. JVT-E146d37ncm, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), November 2002.

[3] ITU-T Recommandation H.263: **Video coding for low bit-rate communication**
Version 1, November 1995; Version 2 (H.263+), January 1998; Version 3 (H.263++), November 2000.

[4] ISO/IEC 14496-2: **Coding of audio-visual objects – Part 2: Visual**.
ISO/IEC JTC1. MPEG-4 Visual version 1, April 1999; Amendment 1 (Version 2), February 2000.

[5] T. Wiegand and B.D. Andrews: **An Improved H.263 Coder Using Rate-Distortion Optimization**
ITU-T/SG16/Q15-D-13, April 1998, Tampere, Finnland.

[6] G.J. Sullivan and T. Wiegand: **Rate-Distortion Optimization for Video Compression**
IEEE Signal Processing Magazine, Vol. 15, November 1998, pp. 74 - 90.

[7] 3GPP TS 26.233 version 5.0.0 Release 5: **End-to-end transparent streaming service; General description**
Universal Mobile Telecommunications System (UMTS), March 2002.

[8] M. Ravasi, M. Mattavelli and C. Clerc: **A Computational Complexity Comparison of MPEG4 and JVT Codecs**
Doc. JVT-D153r1-L, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), July 2002, Klagenfurt, Austria.

[9] VideoLocus Inc.: **AVC Real-Time SD Encoder Demo, July 2002**
Doc. JVT-D023, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), July 2002, Klagenfurt, Austria.

Ralf Schäfer received his Dipl.-Ing. and Dr-Ing. degrees (both in electrical engineering) from the Technical University of Berlin in 1977 and 1984 respectively. In October 1977, he joined the Heinrich-Hertz-Institut (HHI) in Berlin and, since 1989, he has been head of the Image Processing Department where he is responsible for 55 researchers and technicians, about 40 students and about 25 R&D projects. The main R&D fields are Image Processing, Image Coding, Multimedia Communication over (wireless) Internet, Immersive Telepresence Systems and RT-SW implementations and HW design including VLSI.

Dr Schäfer has participated in several European research activities and was chairman of the Task Force on "Digital Terrestrial Television - System Aspects" of the DVB project, which specified the DVB-T standard. Currently, he is a member of the German "Society for Information Technology" (ITG) where he is chairman of the experts committee "TV Technology and Electronic Media" (FA 3.1) and chairman of the experts group "Digital Coding" (FG 3.1.2).

Thomas Wiegand is head of the Image Communication Group in the Image Processing Department of the Heinrich Hertz Institute in Berlin, Germany. He received a Dipl.-Ing. degree in Electrical Engineering from the Technical University of Hamburg-Harburg, Germany, in 1995 and a Dr-Ing. degree from the University of Erlangen-Nuremberg, Germany, in 2000.

From 1993 to 1994, he was a Visiting Researcher at Kobe University, Japan. In 1995, he was a Visiting Scholar at the University of California at Santa Barbara, USA, where he started his research on video compression and transmission. Since then, he has published several conference and journal papers on the subject and has contributed successfully to the ITU-T Video Coding Experts Group (ITU-T SG16 Q.6) standardization efforts. From 1997 to 1998, he has been a Visiting Researcher at Stanford University, USA, and served as a consultant to 8x8 (now Netergy Networks), Inc., Santa Clara, CA, USA.

In October 2000, Dr Wiegand was appointed as Associated Rapporteur of the ITU-T Video Coding Experts Group. In December 2001, he was appointed as Associated Rapporteur / Co-Chair of the Joint Video Team (JVT) that has been created by the ITU-T Video Coding Experts Group and the ISO Moving Pictures Experts Group (ISO/IEC JTC1/SC29/WG11) for finalization of the H.264/AVC video coding standard. He is also the editor of H.264/AVC. His research interests include video compression, communication and signal processing as well as vision and computer graphics.

Heiko Schwarz is with the Image Processing Department of the Heinrich Hertz Institute in Berlin, Germany. He received a Dipl.-Ing. degree from the University of Rostock in 1996 and a Dr-Ing. degree from the University of Rostock in 2000. In 1999, he joined the Heinrich Hertz Institute in Berlin. His research interests include image and video compression, video communication as well as signal processing.

[10] A. Joch, J. In and F. Kossentini: **Demonstration of "FCD-Conformant" Baseline Real-Time Codec**, Doc. JVT-E136, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), October 2002, Geneva, Switzerland.

Acknowledgements

The authors would like to thank Anthony Joch and Faouzi Kossentini for generating and providing the MPEG-2 and H.263+ test results.

Ojo™ Personal Video Phone

The vision is real.

With true-to-life picture and sound the Motorola OJO™ is set to change the face of communication forever.

The implementation of advanced telephony, compression, and multi-media technologies enables OJO to deliver the highest quality images and eliminate the break-up and distortion normally associated with video phones.

BROADBAND FRIENDLY:

Designed for broadband, this system leverages the existing cable and DSL infrastructure while opening the door for new revenue opportunities. As a SIP-compliant endpoint, Ojo gives broadband operators flexibility in the provisioning and administration of a fee-based video telephony service. Ojo requires no additional headend equipment for DOCSIS® cable modems.

BREAKTHROUGH PERSONAL DESIGN:

Stylish and functional. Users can make IP video calls and PSTN or VoIP voice-only calls over a current telephone number. Features include:

- Superior image and bandwidth efficiency
- High-resolution 16:9 LCD display
- State-of-the-art miniature camera
- True-to-life video and audio quality
- Video and voice-only messaging
- Picture-based caller ID and phonebook
- Use of existing telephone number
- Advanced speakerphone with AGC and echo cancellation
- Full-featured cordless phone handset
- An easy-to-use graphical interface
- Easily accessible video/audio privacy controls
- Latest video and audio codecs
- On-screen residential and business directories

WOWMEMOTO

Continued on back

Continued from front

Ojo™ Personal Video Phone

TECHNICAL SPECIFICATIONS

DISPLAY UNIT

GENERAL	
DC Input	12 V
DC Current	3 A (Typical)
Power Consumption	30 W
AC Power Adapter	100 - 240 VAC, 60 Hz
Operating Temperature	10° to 40° C
Storage Temperature	0° to 70° C
Dimensions	14" x 8.5" x 7.5"
Weight	2.5 lb

NETWORK	
Connector	RJ-45
Protocol	TCP/IP
Ethernet Network Interface	10/100 Base-T
Communications Standards	SIP
	TCP/IP, UDP
	RTP
Security	SRTP, 128-bit AES
Call Bandwidth Requirements	110-150 Kbps

PSTN	
Connector	RJ-11
Pass-Through	Yes
Dialing Mode	Tone (DTMF)/Pulse

AUDIO	
Compression (Video Calls)	iLBC
Compression (Audio Calls)	G.711

DISPLAY	
LCD Monitor	Minimum 7" Diagonal
Type	LCD
Backlighting	Yes
Anti-Glare Coating	Yes
Viewing Angle	+/-30°(h) +/-60°(v)

CAMERA	
Image Sensor	1/4" Color
Backlight Compensation	Yes
Automatic Gain Control	Yes
White Balance	Auto
Minimum Illumination	2 lux

SPEAKERPHONE	
Audio Processing	Full Duplex
Echo Cancellation	Adaptive Sub-Band
Audio Privacy	Yes

VIDEO	
Resolution	176 x 144 (QCIF)
Frame Rate	30 fps
Compression (Primary)	H.264
Compression (Supported)	H.263

CORDLESS HANDSET

GENERAL	
Dimensions	6.25" x 1.5" x 0.5"
Weight	5 oz
Wireless Interface Standard: Digital or Analog	2.4 GHz
Range	100 ft
Display	Illuminated Graphic LCM

BATTERY	
Charge Time	1 hr
Talk Time	6 hr
Standby Time	96 hr

**To view our full line of Broadband Products,
visit our Web site at www.motorola.com/broadband/consumers**

MOTOROLA and the Stylized M logo are registered in the US Patent & Trademark Office. WorldGate and Ojo are trademarks of Worldgate Service, Inc. 2004 WorldGate Service, Inc. All other product or service names are property of their respective owners. DOCSIS® is a registered trademark of Cable Television Laboratories, Inc. © Motorola, Inc. 2004

MOTOROLA
intelligence everywhere™

5630-1203-5K